

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

This Page Blank (uspto)



DEUTSCHES
PATENTAMT

US 5 784 636

- 21 Aktenzeichen: 197 22 365.6
22 Anmeldetag: 28. 5. 97
43 Offenlegungstag: 4. 12. 97

30 Unionspriorität:

654395 28.05.96 US

71 Anmelder:

National Semiconductor Corp., Santa Clara, Calif.,
US

74 Vertreter:

Sparing . Röhl . Henseler, 40237 Düsseldorf

72 Erfinder:

Rupp, Charlé R., Bolton, Mass., US

Prüfungsantrag gem. § 44 PatG ist gestellt

54 Rekonfigurierbares Rechenbauelement

- 57 Architektur für Informationsverarbeitungsvorrichtungen, die den Aufbau von kostengünstigen Hochleistungssystemen für spezialisierte Rechenanwendungen einschließlich Sensordatenverarbeitung zuläßt. Die erfindungsgemäße rekonfigurierbare Prozessorarchitektur bedient sich einer adaptiven Logikprozessor (ALP) genannten programmierbaren Logikstruktur. Diese Struktur ähnelt einer erweiterbaren frei-programmierbaren Logikanordnung (FPGA - field programmable gate array) und ist für die Implementierung von programmspezifischen Pipelinefunktionen optimiert, wobei die Funktion beliebige Male während des Verlaufs einer Berechnung geändert werden kann. Eine rekonfigurierbare Pipelinebefehlssteuerungs-(RPIC - Reconfigurable Pipeline Instruction Control) Einheit wird zum Einladen der Pipelinefunktionen in den ALP während des Konfigurationsvorgangs und zum Koordinieren der Operationen des ALPs mit anderen Informationsverarbeitungsstrukturen wie beispielsweise Speicher, E/A-Vorrichtungen und Arithmetikverarbeitungseinheiten benutzt. Es können mehrere Bauelemente mit der rekonfigurierbaren Architektur nach der vorliegenden Erfindung kombiniert werden, um Hochleistungs-Parallelverarbeitungssysteme zu erzeugen, die auf dem Konzept der SIMD-(Single Instruction Multiple Data) Architektur basieren.

DE 197 22 365 A 1

DE 197 22 365 A 1

THIS PAGE BLANK (USPTO)

Ungleichförmigkeit schwieriger. Die Konstruktion ist in einem gleichförmigen Konstruktionsmedium vereinfacht, da die einzigen physischen Beschränkungen, die den Konstruktionsvorgang beeinflussen, die E/A-Stifte sind, die Daten in den und aus dem Baustein bringen. Sobald jedoch feste Schaltungen eingeführt sind, ist nicht nur ein Teil der Funktionalität des Systems festgelegt, sondern es geht auch ein Teil der ursprünglichen Flexibilität der FPGA verloren.

Der Verlust an Flexibilität beeinflusst sowohl Implementierung als auch Leistung. Beispielsweise sind Schaltungsimplementierungen mit mehreren Bausteinen weniger wirksam, da die zur Kommunikation zwischen Bausteinen benutzte chipexterne Verbindung nicht so dicht ist und einen niedrigeren Leistungsgrad aufweist, als chipintern zur Verfügung steht. Schaltungsmodule, die über mehrere FPGA aufgeteilt werden müssen, weisen typischerweise eine geringere Leistung auf, als wenn das Modul in eine einzige FPGA paßt. Partitionierung einer Schaltungsimplementierung über eine Menge fester und unfester Bauelemente ist ebenfalls sehr schwierig.

Wenn beispielsweise bestimmt wird, daß für eine gegebene Anwendung ein Festfunktionsspeicher notwendig ist, müssen alle Schaltungen, die mit dem Speicher kommunizieren, in FPGA mit Stiften, die mit der Multiplizierschaltung verbunden sind, gelegt werden.

Ein weiteres Verbindungsproblem, das die Leistung typischer rekonfigurierbarer Systeme beeinflusst, ist das der Art und Weise, auf die Signale durch die Elemente einer FPGA-Anordnung geführt werden. Signale werden typischerweise vor Verbindung mit einem E/A-Bus oder einem sonstigen Systembauelement zum Rand einer Anordnung geführt. Daraus ergeben sich Verzögerungen bei der Übertragung von Daten und der Ausführung von Logikoperationen. Durch diese Art und Weise der Datenwegeführung wird auch die Flexibilität der rekonfigurierbaren Elemente verringert, da einige Zellen als Drähte konfiguriert werden müssen, damit die Signale durch die Zellen geführt werden können.

Gegenwärtige rekonfigurierbare Systeme weisen bedeutende Schwierigkeiten bei der Implementierung von Anwendungssoftware bei der Platzierung und Wegeführung der rekonfigurierbaren Funktionszellen und rekonfigurierbaren Verdrahtungszellen, die zur Realisierung der rekonfigurierbaren Rechenfunktionsblöcke notwendig sind, auf. Automatische Werkzeuge für dieses Problem benutzen das rekonfigurierbare Mittel allgemein auf unwirksame Weise. In vielen Fällen ist zur Realisierung einfacher rekonfigurierbarer Funktionsblöcke eine umständliche Platzierung und Wegeführung von Hand notwendig.

Die Taktgeschwindigkeitsleistung gegenwärtiger rekonfigurierbarer Systeme ist durch das Erfordernis mehrerer rekonfigurierbarer Drahtbrücken zur Übertragung von Daten aus der Peripherie der rekonfigurierbaren Anordnung zu den inneren Schaltungen, die die Daten benötigen, und die entsprechenden rekonfigurierbaren Drahtbrücken zum Herausbringen der Ergebnisdaten aus den inneren Zellen zur Peripherie der Anordnung begrenzt. Die veränderlichen Wegeverzögerungen des gegenwärtigen Systems erzeugen auch Taktschlupfprobleme bei der Implementierung der Pipelinestufen-Taktsteuerungen.

Gegenwärtige Systeme erfordern, daß alle Pipeline-Datenwege unter Verwendung einer begrenzten Menge an rekonfigurierbaren Verdrahtungsmitteln zur Peripherie der Anordnung gebracht werden. Dadurch wird die Anzahl von Pipeline-Datenbussen begrenzt, die wirkungsvoll in einem System benutzt werden können. Es besteht in gegenwärtigen Systemen eine entsprechende Begrenzung bei der Anzahl von Pipeline-Steuersignalen, die zur Verwaltung des rekonfigurierbaren Pipelinerechenprozesses verteilt werden können.

Erwünscht ist eine Architektur für ein rekonfigurierbares Rechnersystem, mit der die Nachteile bestehender Systeme überwunden werden.

Die gegenwärtige Erfindung richtet sich auf eine Architektur für Informationsverarbeitungsvorrichtungen, die den Aufbau von kostengünstigen leistungsfähigen Systemen für spezialisierte Rechenanwendungen zur Sensordatenverarbeitung erlaubt. Zu einer typischen Anwendung gehört die hochratige Dateneingabe von einem oder mehreren Sensoren, Reduktion der Daten unter Verwendung von aufwendigen Signalverarbeitungs-algorithmen, die Darstellung der Ergebnisse am Systemausgang und die nachfolgende Steuerung von Betätigungsvorrichtungen (beispielsweise den Digital-Analog-Wandlern).

Ein Merkmal der rekonfigurierbaren Rechenarchitektur der vorliegenden Erfindung besteht in der Verwendung einer als adaptiver Logikprozessor (ALP) bezeichneten programmierbaren Logikstruktur. Diese Struktur ist einer erweiterungsfähigen freiprogrammierbaren Logikanordnung (FPGA-field programmable gate array) ähnlich und ist für die Implementierung von anwendungsprogrammspezifischen Pipelinefunktionen optimiert, wobei die Funktion während des Verlaufs eines Rechenvorgangs beliebige Male geändert werden kann.

Diese Pipelinefunktionen werden durch Konfigurierung einer Gruppe von in der Logikanordnung enthaltenen Logikzellen zur Ausführung einer spezifischen Operation bzw. Folge von Logikoperationen implementiert. Zum Beladen des ALP mit den für die Ausführung eines bestimmten Programms benötigten Pipelinefunktion während des Konfigurationsvorgangs wird eine rekonfigurierbare Pipelineanweisungssteuerung(RPIC-Reconfigurable Pipeline Instruction Control) Einheit benutzt. Die RPIC bewirkt auch die Koordinierung der Operationen des ALPs mit anderen Informationsverarbeitungsstrukturen wie beispielsweise Speicher, E/A-Vorrichtungen und Arithmetikverarbeitungseinheiten.

Es können mehrere Bauelemente mit der rekonfigurierbaren Architektur der vorliegenden Erfindung kombiniert werden, um parallele Hochleistungsverarbeitungssysteme zu erzeugen, die auf dem Konzept der SIMD-(Single Instruction Multiple Data)Architektur basieren. In einem solchen Fall wird zur Bereitstellung von Kommunikation zwischen den Bauelementen eine Schaltbus (Toggle Bus) genannte Verbindungsstruktur benutzt. Der Schaltbus wird auch zur Integration von Peripherieverarbeitungselementen wie beispielsweise Bauelementen zum Ablesen von Sensordaten, zur Kommunikation von Ergebnissen und zur Steuerung von Betätigungsgliedern benutzt. Zusätzlich zu den ALP- und PIC-Elementen enthält ein typisches rekonfigurierbares Bauelement eine Schaltbuskoppler-(TBT-Toggle Bus Transceiver)Schaltung, eine Pipeline-Datenprozessor (PDP) genannte standardmäßige arithmetische Pipelineprogrammausführungseinheit, einen Speicher mit Mehrfachzugriff (MPM-Multiple Port Memory) und eine externe Adreßgenerator (XAG) genannte externe Speicher-

zugriffseinheit.

Weitere Aufgaben und Vorteile der vorliegenden Erfindung werden aus der nachfolgenden ausführlichen Beschreibung und den beiliegenden Zeichnungen ersichtlich werden.

Es zeigen:

5 Fig. 1 ein Diagramm der dem rekonfigurierbaren Signalverarbeitungssystem der vorliegenden Erfindung zu Grunde liegenden Rechenverfahren,

Fig. 2 ein Blockschaltbild eines allgemeinen rekonfigurierbaren Signalverarbeitungssystems (RSPS-Reconfigurable Signal Processing System),

Fig. 3 ein Blockschaltbild des Grundaufbaus eines typischen RSPS-Bauelements,

10 Fig. 4 das Format der Grundbefehlsarten für eine beispielhafte PDP- und ALP-Befehlsschnittstelle,

Fig. 5 die Grundelemente eines ALPs bei einer Implementierung mit 32 horizontalen Zellen und 64 vertikalen Zellen,

Fig. 6 die klassische Befehlsausführungsfolge nach von Neumann, die durch den ALP implementiert wird,

15 Fig. 7 die Steuerschnittstelle zwischen den RPIC- und ALP-Modulen und die für jeden Pipeline-Steueranschluß (PCP-pipeline control port) benötigten Signale,

Fig. 8 ein Anwendungspipeline-segment, das eine Addier-/Akkumulierfunktion durchführt,

Fig. 9 die detaillierte Schnittstelle für eine ALP-Kernzelle,

Fig. 10 eine beispielhafte Schaltung zur Implementierung einer ALP-Kernzelle,

Fig. 11 eine beispielhafte Schaltung für die Spaltentakteitung einer ALP-Zelle,

20 Fig. 12 den allgemeinen Plan des Kernblockaufbaus für einen Kernblock mit vier horizontalen Spalten und vier senkrechten Zeilen,

Fig. 13 die Verbindungen, die einen Teil der Schnittstelle zwischen den Kernzellen und Zwischenverstärkerschaltungen an der Ecke von vier Kernblöcken bilden,

Fig. 14 eine beispielhafte Schaltung für eine vertikale Zwischenverstärkerschaltung,

25 Fig. 15 eine beispielhafte Schaltung für eine horizontale Zwischenverstärkerschaltung,

Fig. 16 ein Blockschaltbild mit dem detaillierten Datenfluß für die Hauptbuswege eines beispielhaften RSP-Bauelements,

Fig. 17 ein Blockschaltbild mit einer beispielhaften Schaltung für den Schaltbuskoppler für ein RSP-Bauelement,

30 Fig. 18 ein Blockschaltbild der Grundbauelemente einer RPIC-Schaltung,

Fig. 19 ein Zustandsdiagramm des Betriebsflusses zur Erzeugung eines Pipeline-Freigabesignals für das eingeleitete Programm und selbständige Datenübertragungsoperationen,

Fig. 20 ein Zustandsdiagramm des Betriebsflusses, wenn der Befehlsdecoder bestimmt, daß der gegenwärtige Befehl die Ausführung einer Pipelinefunktion im ALP erfordert,

35 Fig. 21 ein Zustandsdiagramm des Betriebsflusses der Programmdatenanforderungsoperation,

Fig. 22 ein Zustandsdiagramm des Betriebsflusses der selbständigen Anforderungsfolgeoperation,

Fig. 23 das IEEE-32-Bit-Gleitkomma-Datenformat,

Fig. 24 ein Blockschaltbild einer ALP-Schaltung zur Durchführung von beschleunigten Gleitkommaoperationen,

40 Fig. 25 den Datenfluß und Berechnungen des grundlegenden FFT-Algorithmus für $N = 8$,

Fig. 26 ein Blockschaltbild für eine in jedem Datenverarbeitungselement zur Implementierung der bei der FFT-Berechnung benutzten Permutationsoperation benötigte ALP-Schaltung,

Fig. 27, wie der FFT-Algorithmus durch Aufteilen der Datenprozessoren in Paare, die die Grundberechnung durchführen, vereinfacht wird,

45 Fig. 28 ein Blockschaltbild einer ALP-Pipelineschaltung, die die Adresse für die Koeffizienten, die Adresse zum Lesen und Schreiben der Daten berechnet und die Daten für die FFT-Berechnung über den Schaltbus überträgt,

Fig. 29 eine hierarchische Darstellung eines im Speicher gespeicherten Bildes, das bei einer typischen Bildverarbeitungsanwendung benutzt wird,

50 Fig. 30 die Abbildung von Bildpunkten in einem Makroblock für 16 RSP-Datenprozessoren (Numerierung unter Verwendung von Hexadezimaldarstellung) für eine Bildverarbeitungsanwendung auf Grundlage der Architektur der vorliegenden Erfindung,

Fig. 31 das Ordnen von Daten zwischen den Prozessoren, wodurch Zugang der Daten in mehreren nützlichen Adressierbetriebsarten für eine Bildverarbeitungsanwendung ermöglicht wird,

55 Fig. 32 das allgemeine Suchschema für eine Bewegungsoffseterkennungsanwendung auf Grundlage der Architektur der vorliegenden Erfindung,

Fig. 33 den Datenfluß und die Berechnungen des Algorithmus nach Walsh-Hadamard für $N = 8$,

Fig. 34 ein Blockschaltbild einer konfigurierbaren Logikschaltung, die die Berechnung der Fig. 33 implementiert,

60 Fig. 35 einen Teil eines verschiebbaren Pipeline-segments zur Verwendung bei der Ausführung der Berechnung der Fig. 33.

Obwohl die rekonfigurierbare Rechnerarchitektur der vorliegenden Erfindung in verschiedenen Formen, die für spezifische Verwendungszwecke ausgelegt sind, implementiert werden kann, konzentriert sich die folgende Besprechung auf die Verwendung der Architektur für eine Signalverarbeitungsanwendung. Besondere Ausführungsformen der Erfindung erlauben die Veränderung einer Anzahl von Parametern zur Realisierung von Bauelementen, die die Erfordernisse spezifischer Marktsegmente erfüllen:

1) Die Einzelheiten der Implementierung der Kernzellenanordnung und rekonfigurierbaren Verdrahtungs-

zellen auf niedriger Ebene lassen sich verändern, um entweder auf Nachschlagetabellen basierenden rekonfigurierbaren Architekturen oder feinkörnigen rekonfigurierbaren Architekturen Rechnung zu tragen;

2) Die Anzahl von Kernzellen und rekonfigurierbaren Verdrahtungszellen läßt sich verändern, um einen Kompromiß zwischen Bauelementkosten und Anwendungsschaltungsgröße bei gleichzeitig verbesserter Herstellungstechnik zu bieten;

3) Die Schnittstelle zum Speicher, zum Kernprozessor und zur Schnittstellenerweiterung für parallele Verarbeitung kann mit Festfunktionsschaltungen im selben Bauelement mit dem rekonfigurierbaren Logikmittel oder mit externen Bauelementen, die diese Festfunktionseigenschaften enthalten, verbunden sein;

4) Die Bitzahl in den Busworten der Pipelinebusanordnung läßt sich zur Erfüllung der Erfordernisse der Schnittstelle zum Festfunktionsmittel und der Wortlängenerfordernisse bestimmter Anwendungskategorien verändern. Beispielsweise brauchen Bildverarbeitungsanwendungen typischerweise kürzere Worte als Digitalsignalverarbeitungsanwendungen; und

5) Die Anzahl von Pipelinebusanordnungsdatenbussen läßt sich verringern oder erweitern, um einer verringerten bzw. erweiterten Anzahl von Festfunktionsblöcken Rechnung zu tragen.

Der rekonfigurierbare Signalprozessor (RSP) der vorliegenden Erfindung beruht auf einer Rechenarchitektur für Datenverarbeitung, die konfigurierbare Logik- und Parallelverarbeitungs-Verbindungsstrukturen als Teil des Grund-Rechenmechanismus integriert. Mit diesem Ansatz können Algorithmen derart implementiert werden, daß sich eine Leistungsverbesserung in einer ganzen Größenordnung für eine große Reihe von Anwendungen ergibt. Zur gleichen Zeit ermöglicht die Architektur die Verwendung leistungsstark Softwarekompilierkonzepte zur Vereinfachung der Programmentwicklungsaufgaben, die zum Erschließen ihrer hohen Leistungsfähigkeit benötigt werden.

Der Softwarekompilierer für die Architektur bildet wie in Fig. 1 dargestellt die Datenverarbeitungsaufgaben einer Anwendung auf drei Grundrechenverfahren. Diese Rechenverfahren umfassen konfigurierbare Berechnung, skalare Berechnung und parallele Verarbeitung. Die dem RSP zu Grunde liegende Architektur unterstützt die Integrierung dieser Verfahren auf eng verkoppelter Grundlage. Durch Zusammenwirkung der Hardwarestrukturen können Hunderte von Rechenschritten, an denen die skalaren Pipelineprozessoren, die konfigurierbare Logik und die Parallelverarbeitungsmittel beteiligt sind, in einem einzigen Taktzyklus durchgeführt werden.

Durch den RSP wird die Verwendung des konfigurierbaren Logikmittels als Grundbestandteil der Berechnung betont. Durch den Kompilierer wird die höchste Leistung durch Erzeugung anwendungsspezifischer "tiefer Pipeline"-Schaltungen realisiert, die aus der konfigurierbaren Logik gebildet werden. Bei diesem Ansatz werden freiprogrammierbare Logikanordnungen (FPBA) benutzt und die Durchführung vieler elementarer Rechenschritte in einem einzigen Taktzyklus ermöglicht.

Da die konfigurierbare Logikanordnung die Implementierung von willkürlichen Logiknetzen erlaubt, besteht keine Beschränkung der dem Kompilierer für jede Pipelinestufe verfügbaren Art von Rechengrundelement.

Die RSP-Architektur enthält auch das neuartige Konzept einer Pipeline-Busanordnung (PBA-pipelined bus array) als Teil der konfigurierbaren Logikanordnung zur Vereinfachung des Vorgangs automatischer Schaltungserstellung. Dadurch, daß die PBA die direkte Implementierung von Pipelinestrukturen erlaubt, bietet sie auch im Vergleich zu früheren Ansätzen zur Auslegung von rekonfigurierbaren Architekturen eine viel höhere Leistungsfähigkeit. Die Anzahl von in einer einzigen RSP-Vorrichtung implementierten Pipelinestufen ist durch die Größe der konfigurierbaren Logikanordnung begrenzt. In der zu beschreibenden Ausführungsform ist die PBA eine Erweiterung der feinkörnigen konfigurierbaren Logikanordnungsstruktur.

Durch Verwendung der integrierten Parallelverarbeitungs-Verbindungsstruktur der vorliegenden Erfindung lassen sich mehrere RSP-Vorrichtungen in der "Parallelverarbeitungsdimension" kombinieren. Durch Verwendung mehrerer RSP-Bauelemente können unter Verwendung des "Pipelineverkettung" genannten Konzepts tiefe Pipelines erweitert werden. Dies ermöglicht die Ausführung von mehreren Hundert Algorithmusschritten in einem einzigen Taktzyklus. Auch erlaubt die Parallelverarbeitungs-Verbindungsstruktur die Implementierung von herkömmlichen SIMD-(Single Instruction Multiple Data) und MIMD (Multiple Instruction Multiple Data) Parallelverarbeitungsalgorithmen durch den Softwarekompilierer. Die in der zu beschreibenden RSP-Architektur implementierte spezifische Parallelverarbeitungs-Verbindungsstruktur wird als "Schaltbus" bezeichnet. Mit dieser Struktur kann ein Mehrstufen-Verbindungsnetz implementiert werden. Durch Einbau eines vollen Buskopplers in eine RSP-Vorrichtung ermöglicht die Schaltbusstruktur Zugriff auf die meisten Datenflußfähigkeiten eines Kreuzschienennetzes zu einem Preis, der mit dem Einsatz eines herkömmlichen Busses mit drei Zuständen vergleichbar ist. Für die interessierenden Anwendungen wird mit dem Schaltbus echte "skalierbare Berechnung" realisiert: N-mal Bandbreitensteigerung für (N) Verarbeitungselemente.

Anwendungsentwicklung

Die RSP-Architektur erlaubt die Implementierung von Softwareentwicklungswerkzeugen, die die konfigurierbare Logikanordnung auf mehrere unterschiedliche Weisen benutzen. Zusätzlich zu dem Ansatz der tiefen Pipeline kann das konfigurierbare Logikmittel als Mechanismus zur Steigerung des Befehlssatzes benutzt werden. Beispielsweise kann bei einer Anwendung, die die Handhabung einer großen Anzahl von Galois-Feld-Arithmetikgleichungen erfordert, die konfigurierbare Logik zur Implementierung eines Galois-Feld-Arithmetik-Befehlssatzes programmiert werden. Ein Programmierer kann durch Verwendung der "Operandenüberlastungs" Fähigkeit der Programmiersprache C++ auf diese Fähigkeit zugreifen. Für digitalsignalverarbeitungsorientierte Anwendungen kann die konfigurierbare Logik zur Implementierung der komplexen Adreßberechnungen benutzt werden, die zum Zugreifen auf die Anwendungsdatenbank benötigt werden. Der RSP behält die

rekonfigurierbaren Eingabe-/Ausgabefähigkeiten herkömmlicher konfigurierbarer Logikvorrichtungen. Dadurch kann eine Gruppe von RSP-Vorrichtungen direkt mit Betätigungs-/Sensorvorrichtungen verbunden werden, die Datenübertragung mit sehr hoher Bandbreite erfordern, wie beispielsweise Bildwandler und hochratige Kommunikationsvorrichtungen.

Die auf eine bestimmte Anwendung angewandte Mischung von Rechenansätzen ist von einer detaillierten Analyse der Aufgabenstruktur der Anwendung und der Beschaffenheit der kritischen Rechenfolgen in den "inneren Schleifen" des Algorithmus abhängig. Die Abbildung der Anwendung auf die Mehrzahl möglicher verfügbarer Implementierungen unter Verwendung eines RSP-basierenden Systems ist die Aufgabe mit der größten Komplexität im automatischen Kompilierungsvorgang. Nunmehr wird die allgemeine Methode zur Ausführung dieser Aufgabe beschrieben.

Als erstes versucht der Kompilierer immer, 100% des konfigurierbaren Logikmittels zu benutzen. Der RSP enthält hochratige Rekonfigurierungsfähigkeiten, die es dem Anwendungsprogramm ermöglichen, die Berechnungen in der konfigurierbaren Logik mit dem Fortschreiten der Anwendung durch die Abfolge der erforderlichen Aufgaben zu verändern. Beispielsweise kann die konfigurierbare Logik zuerst zur Durchführung von hochratiger Datensammlung benutzt werden. Danach wird die konfigurierbare Logik mehrere Male verändert, um die Daten zu analysieren, und abschließend zur Ausgabe der Ergebnisse der Datenhandhabungen rekonfiguriert.

Der Kompilierer versucht stets, für jede Aufgabe einer Berechnung die tiefste Pipelineanordnung der konfigurierbaren Logik zu realisieren.

Pipelineverkettung zwischen zwei oder mehr RSP-Vorrichtungen über den Schaltbus wird benutzt, wenn Funktionspipelineanordnungen angetroffen werden, die größer sind, als was in das konfigurierbare Logikmittel einer einzigen RSP-Vorrichtung paßt. In Ermangelung der Erkennung einer tiefen Pipelineanordnung für eine spezifische Phase einer Anwendung benutzt der Kompilierer die konfigurierbare Logik als Mechanismus zur Beschleunigung des Befehlssatzes /der Adressenerzeugung. Diese "dynamischen" Benutzungen der konfigurierbaren Logik finden zusätzlich zur "statischen" Benutzung der konfigurierbaren Logik für anwendungsspezifische Eingabe/Ausgabe statt. Der Kompilierer verwaltet die Verwendung des konfigurierbaren Logikmittels auf ähnliche Weise wie die klassischen Verfahren der "globalen Registeroptimierung". Im Gegensatz zu früheren Ansätzen zu einer rekonfigurierbaren Logik, die nur eine Funktion zu einer Zeit aktiv sein lassen, erlauben die mehreren Pipelinesteueranschlüsse (PCP-Pipeline Control Ports) der RSP-Architektur, daß mehrere Pipeline-segmente aktiv bleiben, während irgendeines der Segmente dynamisch rekonfiguriert wird.

Als zweites erreicht der Kompilierer eine verbesserte Leistung durch Verwendung der durch den Schaltbus bereitgestellten Parallelverarbeitungsfähigkeiten. In einer Gruppe von RSP-Vorrichtungen können gleichzeitig mehrere verschiedene Aufgaben (wie beispielsweise Dateneingabe, Datenanalyse und Datenausgabe) durch funktionsmäßige Verteilung dieser Operationen unter den Prozessoren und Verwendung des Schaltbusses für den Aufgabe-Aufgabe-Datenfluß durchgeführt werden. Dieser Rechenstil wird als MIMD-(Multiple Instruction Multiple Data)Verarbeitung bezeichnet. Eine Gruppe von RSP-Prozessoren kann auch unter Verwendung des SIMD-(Single Instruction Multiple Data) Parallelverarbeitungsverfahren zur Beschleunigung eines bestimmten Algorithmus benutzt werden, der enge Datenflußverkopplung erfordert. Beispielsweise können (N) RSP-Prozessoren zusammen zur Beschleunigung der Berechnung des schnellen Fourier-Transformations-(FFT-Fast Fourier Transform)Algorithmus um einen Faktor von N benutzt werden.

Der Kompilierer implementiert den Rest der Anwendung unter Verwendung von Codefolgen für den skalaren Prozessor. Die Programmteile enthalten den Code für die Rekonfiguration des konfigurierbaren Logikmittels. Der Befehlsstrom des skalaren Prozessors ist auch für die höhere Aufgabenfolgensteuerung unter Verwendung der Schaltbustransaktionen und zugehörigen Pipelinesteuermechanismen verantwortlich. Auf den unteren Ebenen der Berechnung wird die Koordination der Operationen zwischen den Funktionseinheiten durch die "selbstständigen Pipeline"-Fähigkeiten der konfigurierbaren Logikanordnung verwaltet, die später beschrieben werden. Abschließend führt der skalare Prozessor alle übrigen Verarbeitungsaufgaben durch, die nicht der konfigurierbaren Logik oder den Parallelverarbeitungsmodulen zugewiesen worden sind.

Ein Aspekt des Verhaltens von RSP-basierenden Systemen besteht darin, daß dasselbe Unterprogramm auf drei verschiedene Weisen ausgeführt werden kann: (1) vollständig in der konfigurierbaren Logik, wenn dieses Mittel an der spezifischen Stelle in der Berechnung nicht anderweitig verwendet wird, (2) teilweise in der konfigurierbaren Logik, wenn nur ein Teil des konfigurierbaren Logikmittels zur Verfügung steht, oder (3) vollständig im skalaren Prozessor, wenn das gesamte konfigurierbare Logikmittel zur Durchführung von Berechnungen eingesetzt wird, die eine bessere Leistung ergeben.

In der Praxis schreitet die Entwicklung einer Anwendung schrittweise voran. In vielen Situationen besteht wahrscheinlich eine vollständige Implementierung für die Anwendung für eine skalare Architektur. In den meisten Fällen kann dieser Code, um eine bessere Leistung zu erzielen, für den RSP rekompiliert werden, wird aber wahrscheinlich nicht die "beste" Leistung bewirken, da die Algorithmusdefinition typischerweise eine Neigung zu dem Skalarsystem des ursprünglichen Ziels enthält. Dies ist eine gut bekannte Erscheinung bei der parallelen Verarbeitung und wird nur durch sachkundige Änderungen des Programms gelöst, mit denen der Kompilierer bessere Arbeit bei der Codeerzeugung leisten kann. Dieser Weg der "progressiven Verfeinerung" ist eine der Hauptstärken der RSP-Architektur im Vergleich zu früheren Methoden. Das heißt, ein unerfahrener Benutzer wird ohne bedeutenden Aufwand eine nützliche Lösung für ein Anwendungsprogramm erhalten. Mit zunehmenden Kenntnissen des Benutzers können progressiv bessere Lösungen erzielt werden. Ausgeklügelte Softwarewerkzeuge wie beispielsweise "Anwendungsprofilierer" und auf Graphik basierende "Datenflußmanager" werden typischerweise zur Realisierung eines Endlösungsansatzes unter Verwendung der bei der Architektur der vorliegenden Erfindung verfügbaren Rechenverfahren eingesetzt.

Strukturell besteht die Ausgabe eines RSP-Kompiliererwerkzeugs aus einer Objektdatei, die sowohl die

binäre Codedefinition, die skalaren Prozessoren und eine Menge verschiedener Schaltungsstrukturen für die konfigurierbaren Logikanordnungen enthält. Die RSP-Architektur ist darin einmalig, daß sie die automatische Erzeugung von verschiebbaren Logikblöcken auf Grundlage der Schaltungsstrukturen unter Verwendung des Pipeline-Busanordnungskonzepts erlaubt.

5

Übersicht über die RSPS-Systemarchitektur

Wie schon besprochen, basiert das rekonfigurierbare Signalverarbeitungssystem (RSPS-Reconfigurable Signal Processing System) der vorliegenden Erfindung auf einer Architektur für eine Familie von Informationsverarbeitungsvorrichtungen, die zum Aufbau von Hochleistungssystemen für spezialisierte Rechenanwendungen für Sensordatenverarbeitung benutzt werden können. Eine typische Anwendung würde eine hochratige Dateneingabe von einem oder mehreren Sensoren (wie beispielsweise eine Videokamera, einem Radar-Responder, Analog-Digital-Wandler usw.), Reduktion der Daten unter Verwendung komplexer Algorithmen wie beispielsweise der Fourier-Transformation, Darstellung der Ergebnisse am Ausgang des Systems und nachfolgende Steuerung einer Betätigungsvorrichtung (wie beispielsweise einem Digital-Analog-Wandler usw.) umfassen. Ein einmaliges Merkmal von RSPS-Automaten ist die Verwendung einer fortschrittlichen Art von programmierbarer Logikstruktur, die als adaptiver Logikprozessor (ALP – Adaptive Logic Processor) bezeichnet wird. Diese Struktur ist für die Implementierung von programmspezifischen Pipelinefunktionen optimiert, wobei die Funktion beliebige Male während des Verlaufs der Berechnung geändert werden kann. Die neuartige rekonfigurierbare Pipelinebefehlssteuerungs-(RPIC – Reconfigurable Pipeline Instruction Control) Einheit der vorliegenden Erfindung ist für das Einladen der Funktionen in den ALP (eine als Konfigurationsvorgang bezeichnete Operation) und das Koordinieren der Operationen des ALPs mit anderen Informationsverarbeitungsstrukturen verantwortlich. Zu diesen können Speicherelemente, E/A-Vorrichtungen und Arithmetikverarbeitungseinheiten gehören. Wie schon bemerkt, können mehrere RSPS-Architekturbaulemente kombiniert werden, um Parallelverarbeitungssysteme sehr hoher Leistung zu erzeugen, die auf dem Konzept der SIMD-(Single Instruction Multiple Data) Architektur basieren.

Die Fig. 2 ist ein Blockschaltbild eines allgemeinen RSPS (Reconfigurable Signal Processing System) 100. In Fig. 2 wird jedes RSPS-Architekturbaulement ein rekonfigurierbarer Signalprozessor (RSP) 102 genannt. N der Bauelemente, die von 0 bis N-1 nummeriert sind, werden als die Datenverarbeitungselemente des Systems bezeichnet und ein zusätzliches (N) nummeriertes RSP-Baulement 104 wird als Bushauptprozessor bezeichnet. Dieses Element wird als die Steuerung für die Datenelementanordnung benutzt. Die Tatsache, daß dasselbe RSP-Baulement für diese verschiedenen Rollen benutzt werden kann, ist eine direkte Folge der ALP-Rekonfigurationsfähigkeit.

Wie schon bemerkt, wird zur Bereitstellung von Kommunikation zwischen den N RSP-Baulementen eine als Schaltbus 106 bezeichnete Verbindungsstruktur benutzt. Der Schaltbus 106 hat die Eigenschaft, daß die Netto-Signalbandbreite des Busses im Verhältnis zur Anzahl verbundener RSP-Baulemente steigt. Der Schaltbus 106 wird auch zur Integrierung von Peripherie-Verarbeitungselementen wie beispielsweise Bauelementen zum Ablesen von Sensordaten, der Kommunikation von Ergebnissen und der Steuerung von Betätigungsgliedern benutzt. Weitere Informationen hinsichtlich der Struktur und Funktionsweise des Schaltbusses 106 sind aus der am 5. Dezember 1995 eingereichten allgemein zugewiesenen US-Patentanmeldung Seriennummer 08/567 172 mit der Bezeichnung "Schaltbusschaltung" ersichtlich, deren Inhalt hier durch Bezugnahme aufgenommen wird. Jede RSP-Vorrichtung 102 und 104 enthält Speichermittel, die wie in der Figur gezeigt unter Verwendung externer Speicher-Baulemente 108 erweitert werden können. Wenn gewünscht, können Peripherieprozessoren 110 an den Schaltbus 106 angeschaltet werden, um bei Dateneingabe/-Ausgabefunktionen behilflich zu sein.

Struktur zur Implementierung von RSP-Elementen

Der RSP (und das RSPS-System) können als spezialisierte Großrechnerarchitektur betrachtet werden, mit der Betonung auf Anwendungen, die bedeutende Mengen von Handhabungen auf Bitebene und ganzzahlige Arithmetik erfordern. Zu Anwendungen, die in diese Kategorie fallen, gehört Bildverarbeitung und neue Arithmetik wie beispielsweise Endlichfeldarithmetik-Anwendungen. Die Architektur eignet sich auch für Datenbankverarbeitung und Verarbeitung von hochratiger Kommunikation. Die zu beschreibende Rechenarchitektur ist zur Begünstigung der Verwendung des konfigurierbaren Logikmittels ausgelegt. Als solches ist die skalare Verarbeitungsfähigkeit ein ganzzahliger Digitalsignalverarbeitungs- (DSP-Digital Signal Processing) Kern, der mit den Anwendungsarten übereinstimmt, die beschrieben werden. Alternative Ausführungsformen der Architektur könnten Prozessoren mit vollem Gleit-komma und virtuellen Speicher-Adressierungsfähigkeiten enthalten.

Die Fig. 3 ist ein Blockschaltbild des Grundaufbaus eines typischen RSPS-Baulements (RSP-Elemente 102 oder 104 der Fig. 2). Zusätzlich zu den bereits erwähnten Schaltungen des ALP 120 und RPIC 122 enthält eine typische RSP-Vorrichtung eine Schaltbuskoppler-(TBT-Toggle Bus Transceiver) Schaltung 124, eine als Pipeline-Datenprozessor (PDP-Pipeline Data Processor) 126 bezeichnete Pipelineeinheit zur Ausführung von Standardarithmetik, einen Speicher mit Mehrfachzugriff (MPM-Multiple Port Memory) 128 und eine als externer Adreßgenerator (XAG-External Address Generator) 130 bezeichnete externe Speicherzugriffseinheit.

Die Hauptmechanismen zur Implementierung einer RSP-Architektur sind die Pipelinebusanordnung 132 (PBA, in der Figur als Busleitungen RA, WA, RD, WD, RX und WX gezeigt), die rekonfigurierbare Pipelinebefehlssteuerung (RPIC-Reconfigurable Pipeline Instruction Controller) 122 und die Pipelinesteueranschlüsse (PCP-Pipeline Control Ports) 123. Im allgemeinen Fall enthält jedes RSP-Baulement auch das hochratige Speichermittel mit Mehrfachzugriff (MPM-Multiple Port Memory) 128, das gleichzeitigen Zugriff von Programmanweisungen, Datenlesen und Datens Schreiben in einem einzigen Taktzyklus erlaubt. Die dem RSP zur

Verfügung stehende Speicherkapazität kann unter Verwendung von externen Bauelementen erweitert werden, auf die unter Verwendung des externen Adreßgenerators (XAG) 130 und den in der Figur gezeigten externen Verbindungen "A" und "B" zugegriffen wird. Für die Zwecke der gegenwärtigen Besprechung dient der adaptive Logikprozessor (ALP) 120 als das konfigurierbare Logikmittel, der Pipeline-Datenprozessor (PDP) 126 fungiert als der DSP-Kernprozessor und der Schaltbuskoppler (TBT-Toggle Bus Transceiver) 124 dient als die eingebettete Schaltbusverbindungserschaltung. In einer allgemeineren Architektur können die Blöcke PDP und TBT durch andere Arten von skalaren Verarbeitungseinheiten und andere Verbindungsarchitekturstrukturen ersetzt werden. Bei einigen Ausführungsformen können die Blöcke TBT, PDP, MPM und XAG entfernt werden, was nur das konfigurierbare Logikmittel und die Pipelinesteuerfähigkeit als Grundmodule der Architektur beläßt.

Wie bemerkt, ist ein einmaliges Merkmal von RSPS-Bauelementen die Verwendung einer Pipelinebusanordnungs-(PBA-)Struktur 132, die erlaubt, daß beliebige der Funktionseinheiten mit den im ALP 120 konfigurierten Pipelinefunktionen verbunden werden können. Daraus ergibt sich ein schnelleres und wirkungsvolleres Mittel zur Datenübertragung zwischen dem ALP und den anderen Bauelementen. Gleichzeitig können mehrere funktionsmäßig unterschiedliche Pipelinesegmente im ALP benutzt werden, was eine selbständige Pipelineaktion zuläßt, die durch Programmausführung im RPIC 122 synchronisiert wird.

Nunmehr werden die allgemeinen Konzepte für die Elemente ALP 120 und RPIC 122 in RSPS-Bauelementen und die Integration des Schaltbusses 106 besprochen. Wie schon erwähnt, wird angenommen, daß der PDP 126 eine typische hochratige Pipeline-Arithmetikeinheit wie beispielsweise der von National Semiconductor Corporation, dem Inhaber der vorliegenden Erfindung, hergestellte Digitalsignalverarbeitungs(DSP-Digital Signal processing)Kern ist. Es ist zu bemerken, daß die RSPS-Architektur die Verwendung einer Vielzahl von PDP-Funktionseinheiten zuläßt und RSP-Bauelemente berücksichtigt, die keine PDP-Einheit aufweisen. Bei den Einheiten MPM 128 und XAG 130 kommen Grundsätze typischer hochratiger Speicher mit verschachteltem Cache zur Anwendung und sie werden hier nicht ausführlich beschrieben.

In einer Ausführungsform ist ALP 120 eine Erweiterung der FPGA-Architektur des CLAY von National Semiconductor und bewahrt die feinkörnige symmetrische Zellenstruktur dieser Vorrichtung. Die CLAY-Architektur einer konfigurierbaren Logikanordnung ist in dem am 29. März 1994 erteilten und dem Inhaber der vorliegenden Erfindung zugewiesenen US-Patent Nr. 5 298 805 mit der Bezeichnung "Versatile and Efficient Cell-to-Local Bus Interface in a Configurable Logic Array" (Vielseitige und wirkungsvolle Zellen-Lokalbus-Schnittstelle in einer konfigurierbaren Logikanordnung) beschrieben und dessen Inhalt wird hiermit durch Bezugnahme aufgenommen. Die Hauptveränderungen an der CLAY-Architektur betreffen Optimierungen für eine wirkungsvolle Implementierung von mehrstufigen Pipelinestrukturen, Steuerung von mehrfachen Pipelineketten und Zugriff auf die Pipeline-Datenbusse des RSPs. Andere konfigurierbare Logikmittel mit der angegebenen Funktionalität können ebenfalls benutzt werden.

Beschreibung des allgemeinen RSP-Blockschaltbildes

Nach der Fig. 3 besteht ein RSP-Bauelement typischerweise aus sechs Funktionseinheiten:

ALP: Adaptiver Logikprozessor. Eine programmierbare Logikstruktur, die die Implementierung von anwendungsspezifischen Logikschaltungen zur Steuerung externer Bauelemente und Berechnungspipelines erlaubt. Der ALP besteht aus einer Anordnung von Logikzellen, programmierbaren schnittstellen zu den Eingangs-/Ausgangsstiften einer RSP-Vorrichtung, programmierbare Verbindungen zwischen den Logikzellen und Zugriff zu den Daten der Pipelinebusanordnung (PBA). Der gesamte ALP kann der Implementierung einer Funktion zugeordnet sein oder es können mehrere kleinere Funktionen gleichzeitig ablaufen. Die ALP-Schaltungen werden von der RPIC unter Verwendung von Daten im MPM oder vom Schaltbus konfiguriert. Die Schaltungen im ALP werden während der Ausführung eines typischen Programms allgemein mehrere Male geändert. Schaltungen im ALP, bei denen die Konfiguration augenblicklich nicht geändert wird, bleiben aktiv.

RPIC: Rekonfigurierbare Pipelinebefehlssteuerung (Reconfigurable Pipeline Instruction Controller). Verantwortlich für die Koordinierung des Betriebes zwischen Funktionseinheiten. Die RPIC erzeugt den Befehlsadreßstrom zur Programmausführung, decodiert die für jeden Befehl erforderliche Operation, erzeugt Adressen zum Lesen und Schreiben von Daten, erzeugt Steuersignale für den ALP und PDP zur Befehlsausführung und steuert die Rekonfiguration des gesamten ALPs oder eines Teils desselben. Die Steuerschnittstelle zwischen RPIC und ALP benutzt ein Pipelinesteueranschlüsse (PCP-Pipeline Control Ports) genanntes Konzept, das ausführlich unten beschrieben wird.

PDP: Pipeline-Datenprozessor (Pipeline Data Processor). Dies ist eine herkömmliche Pipeline-Arithmetikverarbeitungseinheit. Typischerweise enthält sie eine ganzzahlige Mehrfunktions-Arithmetik Logikeinheit (ALU-Arithmetic Logic Unit). Sie kann auch fest zugeordnete Funktionseinheiten für Multiplikations- und Gleitkommaoperationen enthalten.

Der PDP empfängt vom MPM, TBT oder ALP eingegebene Daten.

MPM: Speicher mit Mehrfachzugriff (Multiple Port Memory). Eine schnelle Speichereinheit, die gleichzeitiges Programmlesen-Datenlesen und Datenschreiben erlaubt. Bei kleinen Anwendungen liefert der MPM alle Erfordernisse des Direktzugriffspeichers (RAM-Random Access Memory). Bei größeren Anwendungen wird der MPM als cachespeicher für außerhalb des RSPs liegende größere Speicher benutzt. Der MPM kann über den Schaltbus oder über einen fest zugeordneten Schnittstellenanschluß für einen externen Speicher beladen werden.

TBT: Schaltbuskoppler (Toggle Bus Transceiver). Dieser stellt einen flexiblen Datenfluß zwischen mehreren RSP-Bauelementen bereit, wenn er in der SIXD-Konfiguration benutzt wird. Der TBT stellt auch eine standardmäßige Eingabe-/Ausgabeschnittstelle für externe Speicher-, Sensor- und Betätigungsbaulemente bereit. In einer Anordnung von N RSP-Bauelementen bilden die TBT-Schaltungen zusammen ein Mehrstufenverbin-

dungsnetz (MIN-Multiple Stage Interconnect Network).

XAG: Externer Adreßgenerator. Damit werden die Adressierungsfähigkeiten des RSP für große chipexterne Speicher erweitert. Auch liefert der XAG die Adreßfolge für Einschaltkonfigurierung des ALPs von einer chipinternen oder chipexternen Nurlesespeicher-(ROM-Read Only Memory)Vorrichtung.

PBA: Pipeline-Busanordnung (Pipeline Bus Array). Acht Sätze von Datenleitungen für Datenflüsse zwischen den Funktionseinheiten des RSPs. Die Aktivität des PBAs erlaubt gleichzeitige Operation der angeschlossenen Funktionseinheiten mit hoher Geschwindigkeit. Der allgemeine Gebrauch jeder PBA-Busdatenleistung ist wie folgt:

PA: Programmadreßbus (Program Address Bus). Dieser definiert die Adresse des nächsten aus dem MBM auszulesenden Befehls. Der PA wird nur von der RPIC zur Programmfolgesteuerung gesteuert und getrieben.

RP: Leseprogrammibus (Read Program Bus). Der MPM-Ausgang liefert Befehlsdaten zur RPIC zur Programmausführung. Der RP kann auch vom TBT zur Ausführung von über den Schaltbus gelieferten Befehlen angesteuert werden.

RA: Leseadreßbus (Read Address Bus). Dieser liefert die Adresse zum Auslesen von Daten aus dem MPM. Die RPIC steuert den RA zum Zugreifen auf durch das Adreßfeld in einem Befehl angegebene Daten an. Der RA kann auch von einer ALP-Pipeline angesteuert werden, um Speicherzugriff für Datenanordnungen zu vereinfachen. Vom ALP erzeugte RA-Werte werden auf zweierlei Weisen von der RPIC synchronisiert. Wenn die Daten, auf die Bezug genommen wird, für den PDP bestimmt sind, dann wird eine konfigurierbare Adreßmodenerweiterung (CAMX-Configurable Address Mode Extension) benutzt. Wenn die Daten für den ALP bestimmt sind, synchronisiert die RPIC den RA mit dem RD-Busgebrauch.

WA: Schreibadreßbus (Write Address Bus). Dieser liefert die Adresse zum Einschreiben von Daten in den MPM. Die RPIC steuert den WA zum Einschreiben von durch das Adreßfeld in einem Befehl angegebenen Daten an. Der WA kann auch von einer ALP-Pipeline angesteuert werden, um Speicherzugriff für Datenanordnungen zu vereinfachen. Vom ALP erzeugte WA-Werte werden auf zweierlei Weise von der RPIC synchronisiert. Wenn die Daten, auf die Bezug genommen wird, für den PDP bestimmt sind, dann wird eine konfigurierbare Adreßmodenerweiterung (CANX-Configurable Address Mode Extension) benutzt. Wenn die Daten für den ALP bestimmt sind, synchronisiert die RPIC den WA mit dem WD-Busgebrauch.

RD: Lesendatenbus (Read Data Bus). Dieser Bus führt Daten vom MPM für den PDP, Daten vom MPM für den ALP oder Daten vom ALP für den PDP. Datenübertragung vom ALP zum PDP ist stets mit einer CAMX synchronisiert.

WD: Schreibdatenbus (Write Data Bus). Dieser Bus führt Daten vom PDP zum Einschreiben in den MPM, Daten vom ALP zum Einschreiben in den MPM oder Daten vom PDP zum ALP. PDP-ALP-Übertragungen werden unter Verwendung einer CAMX synchronisiert. ALP-MPM-Übertragungen werden durch Befehle "ALP speichern" oder durch eine unten beschriebene Folge selbständiger Pipelineaktion (APA-Autonomous Pipeline Action) synchronisiert.

RX: Extern-Lesebus (Read External Bus). Dieser Bus führt chipexterne Daten vom TBT zum PDP, ALP oder MPM. Das Lesen von externen Daten wird durch die RPIC synchronisiert. TBT-ALP-Übertragungen speisen Daten in eine ALP-Pipeline, die mit anderen Orten verkettet sein kann. TBT-PDP-Datenübertragung wird durch Ausführung von Befehlen "IN" eingeleitet. TBT-ALP-Übertragungen werden durch Befehle "IN" mit CANX oder durch selbständige Pipelineaktion eingeleitet.

WX: Extern-Schreibbus (Write External Bus). Dieser Bus führt vom PDP oder ALP ausgegebene Daten über den Schaltbus zu externen Bauelementen. PDP-Ausgabe wird unter Verwendung von Befehlen "OUT" eingeleitet. ALP-TBT-Übertragungen werden durch Befehle "OUT" oder durch selbständige Pipelineaktion eingeleitet. Zusätzlich zu den PBA-Bussen enthält ein typisches RSP-Bauelement weitere spezialisierte Daten- und Steuerbusse 131. Es gibt fünf Hauptdatenwege, die die Verbindung eines RSP-Bauelements mit anderen RSP-Bauelementen, chipexternen Speichern, Sensoren und Betätigungsgliedern ermöglichen. Die allgemeine Nutzung dieser Datenwege ist wie folgt:

P: Haupt-Schaltbusschnittstellenbus (Primary Toggle Bus interface bus). Dies ist die vorgegebene Datenschnittstelle für einen externen Speicher. Wenn mehrere RSP-Bauelemente in einer SIMD-Anordnung benutzt werden, arbeiten die P- und Q-Busse zusammen, um Parallelverarbeitungsdatenflußmuster zu erstellen. Wenn nur ein RSP in einem System benutzt wird, fungiert der P-Bus als standardmäßiger Bus mit drei Zuständen (siehe Fig. 17).

Q: Sekundärer Schaltbusschnittstellenbus (Secondary Toggle Bus interface bus). Wenn nur ein RSP in einem System benutzt wird, kann der Q-Bus als zweiter Bus mit drei Zuständen oder als rekonfigurierbarer Satz von Eingangs-/Ausgangsstufen für externe Speicher-, Sensor- und Betätigungsgliederverbindung benutzt werden. Wenn mehrere RSP-Bauelemente in einer SIMD-Anordnung benutzt werden, werden die Q- und P-Leitungen unter Verwendung eines "Mischverdrahtungs"-Musters zusammengeschaltet. P und Q arbeiten dann zusammen mit dem Satz TBT-Schaltungen, um ein MIN-Parallelverarbeitungs-Verbindungsnetz zu erstellen (siehe Fig. 17).

A: Primärer Adreßbus (Primary Address Bus). Als Vorgabe wird der A-Bus zur Adressierung von externen Speicherbauelementen von der XAG-Einheit aus benutzt. Wenn er nicht für diesen Zweck benutzt wird, kann der A-Bus durch programmierbare Logik im ALP als rekonfigurierbare Eingabe-/Ausgabeleitungen zum Anschalten an externe Sensoren und Betätigungsglieder angesteuert werden.

C: Steuerbus (Control Bus). Wenn mehrere RSP-Bauelemente in einer SIMD-Anordnung benutzt werden, bildet der C-Bus den Schaltbussteuervektor, der vom Bushauptsteuerungs-RSP (Bauelement N der Fig. 2) erzeugt wird. In diesem Fall ist C ein Eingangsvektor zu den Datenverarbeitungs-RSP-Bauelementen (Bauelemente 0 bis N-1). Wenn nur ein RSP in einem System benutzt wird, kann der C-Bus durch programmierbare Logik im ALP zur Steuerung externer Sensoren und Betätigungsglieder angesteuert werden (siehe Fig. 17).

B: Sekundärer Datenbus (Secondary Bus). Als Vorgabe ist der B-Bus eine sekundäre zweiseitig gerichtete

Schnittstelle für externe Speicherbauelemente unter der Steuerung des XAGs. In dieser Betriebsweise können externe Daten zwischen dem RSP und externen Speichern ohne Verwendung des Schaltbusses übertragen werden. Dies ermöglicht eine höhere Leistung bei Anwendungen, die die gesamte Schaltbusbandbreite für Prozessor-Prozessor-Kommunikation benutzen. Der B-Bus kann vom ALP als rekonfigurierbare Eingabe-/Ausgabelleitungen für den Anschluß von externen Sensoren und Betätigungsgliedern angesteuert werden.

Zusätzlich zu diesen primären Daten und Adreßschnittstellenbussen sind mehrere einzelne Steuersignalschlüsse an einem RSP-Baustein der Takteingabe, externen Speichersteuerung usw. zugeordnet.

Datenformate und Speicheradressierungsmodi

Ein RSP-Bauelement kann unter Verwendung von 16-Bit-, 32-Bit- oder 64-Bit-Datenwegen entwickelt werden. Für Darstellungszwecke wird ein RSP-Bauelement mit 16-Bit-Datenworten benutzt. Jede Speicheradresse bezieht sich dann auf ein 16-Bit-Datenwort. Genaue Einzelheiten der Befehlswortbit werden von der Art der benutzten PDP-Einheit abhängig sein. Die Fig. 4 zeigt das Format der Grundbefehlstypen für eine beispielhafte PDP- und ALP-Befehlsschnittstelle. Der Befehl des Typs Nulladresse (ZA-zero address) benutzt ein gesamtes Wort für den Operationscode (OP) des Befehls und wird für Operationen benutzt, die keine Befehlsad- oder Parameterfelder (beispielsweise Löschen Akkumulator, Beginnen Konfigurationsfolge) erfordern. Die Befehle mit dem Format kurze Konstante (SC-short constant) benutzen einen 8-Bit-Operationscode und ein 8-Bit-Konstantenfeld (C). SC-Befehle führen Funktionen wie beispielsweise Hinzuzaddieren einer Konstante zu einem Akkumulatorergebnis durch. Das Format lange Konstante (LC-long constant) benutzt zwei Worte, um Operationen mit einer vollständigen 16-Bit-Konstante zu ermöglichen. Die meisten Arithmetikbefehle im RSP sind von dem Format Kurzspeicher (SM-short memory). Die oberen acht Bit definieren den Operationscode. Das höchstwertige Bit des unteren Bytes (i) ist das Steuerbit "indirekte Speicherreferenz" und die übrigen sieben Bit (A) definieren einen kurzen Adreßwert oder einen indirekten Speicherzugriffsmodus, wie unten beschrieben. Das Langspeicherformat (LM-long memory) erlaubt direkten Zugriff zu dem vollen Speicherbereich einschließlich chipexterner Speicherreferenz. Das LM-Format wird auch für Befehle des Typs "Sprung" und "Aufruf" benutzt. Das Speicherverschiebe-(MS-memory shift)Format erlaubt Speicherlese- und Schreiboperationen mit zugehöriger Datenverschiebung unter Verwendung eines Vier-Bit-Verschiebungszählfeldes (S). Die Speicherregisterbefehle (MR-memory register) ermöglichen die Übertragung von Daten zwischen dem Speicher und bis zu acht Adreßregistern in der RPIC.

Die Adreßwerte, auf die durch RSP-Befehle und die PA-, RA- und WA-Busse Bezug genommen wird, zeigen auf Werte in vier getrennten Adreßräumen in Abhängigkeit vom Befehlstyp:

Programmspeicher — Ziel von Befehlen des Typs "Sprung" und "Aufruf";

Datenspeicher — Ziel von ALP- und PDP-Datenlese- und Schreibbefehlen wie beispielsweise "Zuaddieren zum Akkumulator";

E/A-Raum — Ziel von externen ALP- und PDP-Datenübertragungen. Wird durch die Befehle des Typs "in" und "out" benutzt;

Konfigurierungsraum — Werte der programmierbaren ALP-Kernzellen- und Verbindungsfunktion. Die zur Definierung der ALP-Schaltungen erforderlichen Daten sind in einem linearen Adreßraum zur Änderung durch das Programm adressierbar. Ziel der Konfigurations-Lese- und Schreibbefehle.

Auf den Datenspeicherraum wird direkt im RSP zugegriffen, wenn das indirekte Adressierungsbit des Befehls einen Wert "0" aufweist. In diesem Fall wird das sieben-Bit-Adreßfeld mit einem "Datenseite"-Adreßwert in der RPIC verkettet, um die vollständige Datenspeicheradresse zu bilden.

Der Wert "Datenseite" kann durch einen Befehl "Seite setzen" geändert werden. Die meisten Datenspeicherbezugnahmen werden unter Verwendung indirekter Adressierung (i Bit ist "1") durchgeführt. In diesem Fall beziehen sich die niedrigeren drei Bit des Adreßfeldes (Feld R) auf das zu benutzende Adreßregister in der RPIC und die übrigen vier Bit definieren den Adressierungsmodus (AM) nach Tabelle 1. R0 stellt den Inhalt des Adreßregisters 0 dar.

Tabelle 1

RSP-Befehlsadreibmodi

AM	Name	Daten
0	Register Direkt	(R) Inhalt des Registers
1	Register Indirekt	MPM [(R)] Register ist Adresse der Daten im MPM-Datenraum
2	Automatische Erhöhung	MPM [(R++)] Erhöhen R nach Zugriff
3	Automatische Erniedrigung	MPM [(--R)] Erniedrigen R vor Zugriff
4	Indexiert Direkt	(R) + (RO)
5	Indexiert Indirekt	MPM [(R) + (RO)]
8	ALPA	Lesen: MPM [RA]; ALP-Pipeline liefert Leseadresse Schreiben: MPM [WA]; ALP-Pipeline liefert Schreibadresse
9	ALPW	WD; PDP-Ausgabe liefert Daten zu ALP-Pipeline(s)
10	ALPRO	RD; ALP-Pipeline 0 liefert Daten zum PDP
11	ALPRI	RD; ALP-Pipeline 1 liefert Daten zum PDP

Adreibmodi 0 bis 5 stellen typische Adressierungsmodi dar und sind von der genauen Struktur des PDP-Moduls abhängig. Adreibmodi 8 bis 11 sind einmalig für das RSP-Konzept der vorliegenden Erfindung und werden als konfigurierbare Adreibmodi (CAM-Configurable Address Modes) bezeichnet. Der ALP-Modus erlaubt, daß die Adresse für eine Speicherschreib- oder Leseoperation von einer "RA-Pipeline" im ALP zum Lesen und durch eine "WA-Pipeline" im ALP zum Datens Schreiben geliefert wird. Die Adreibmodi ALPRO und ALPRI erlauben die direkte Verwendung der Ausgabe von einer von zwei ALP-Pipelines als Operand für eine Operation im PDP. Der Adreibmodus ALPW erlaubt die direkte Verwendung von Daten aus dem PDP als Operandeneingabe in eine oder mehrere ALP-Pipelines. Zusätzlich zum Zugriff auf die Pipeline-Busanordnung (PBA) im ALP durch konfigurierbare Adreibmodenoperationen kann durch M-PM-ALP- und ALP-MPM-Übertragungsbefehle explizit auf ALP-Daten Bezug genommen werden. Datenübertragungen können auch unter Verwendung der selbstständigen Pipelinesteuerungs-(APC-Autonomous Pipeline Control) Fähigkeit auftreten, die unten erklärt wird. Es ist zu bemerken, daß, obwohl die Tabelle 1 eine mögliche Menge von Registerzuweisungen zeigt, andere Zuweisungen in Abhängigkeit von den Erfordernissen des ALP-Kerns benutzt werden können.

Allgemeine Struktur des adaptiven Logikprozessors

Der adaptive Logikprozessor (ALP-Adaptive Logic Processor) besteht aus einer Anordnung programmierbarer Logikzellen und einer Anordnung programmierbarer Schalter, die den Datenfluß zwischen den Zellen erlauben. Diese Elemente bilden ein konfigurierbares Logikmittel, das in die RSP-Architektur eingebaut ist. Der ALP enthält die pipeline-Busanordnung (PBA) und rekonfigurierbare Eingangs-/Ausgangsanschlüsse. Die Fig. 5 stellt die Grundelemente des ALP 120 in einer Ausführungsweise mit 32 horizontalen Zellen und 64 vertikalen Zellen dar.

Im allgemeinen Fall besteht ALP 120 aus drei Schaltungsschichten, die für die RSP-Architektur spezifische Funktionen durchführen. Die Grundschrift wird die "Kernzellenanordnung" genannt und besteht aus einer zweidimensionalen Anordnung von logischen Kernzellenschaltungen 150. Jede Kernzellenschaltung enthält ein Konfigurationsregister, das unter Steuerung des RPIC-Blocks beladen werden kann, wenn ein Befehl "Rekonfiguration" ausgeführt wird. Der Wert im Konfigurationsregister definiert die von der Kernzelle durchzuführende Logikfunktion und definiert Verbindungen für die Logikeingabe- und Ausgabesignale der Kernzelle. Die konfi-

gurierte Funktion einer Kernzelle ist typischerweise eine Boolesche Operation mit mehreren Eingaben und mehreren Ausgaben, die einen oder mehrere Daten-Flipflops enthalten kann. Bei einer typischen Ausführungsform enthalten einige oder alle der zur Implementierung der Booleschen Operation benötigten Signale Verbindungen zu den nächsten Nachbarkernzellen. Dies stellt die unterste Ebene von Kernzellenverbindungen dar.

Die zweite logische Schicht der konfigurierbaren Logikanordnung (ALP) besteht aus Signalleitungen und Schaltern, die die Übertragung von Signalwerten über eine Entfernung von mehreren Kernzellen erlauben. Dies wird als "Lokalbusanordnung" bezeichnet. Jede Kernzelle kann eine Eingabe von einem oder mehreren Lokalbusanordnungsverdrahtungssegmenten empfangen und kann Daten zu einem oder mehreren Lokalbusanordnungssegmenten liefern. Jeder Satz von Lokalbusverdrahtungswegen überspannt einen als "Kernblock" 152 in der Figur bezeichneten Bereich. Die Schaltungen, die einen Signalaustausch zwischen Kernblöcken erlauben, werden (in der Figur nicht gezeigte) "Verstärker-Schaltungen" genannt. Jede Verstärkerschaltung enthält auch ein Konfigurationsregister, das definiert, wie Signale von einem Kernblock zu einem anderen fließen. Die zweite logische Schicht der konfigurierbaren Logikanordnung enthält auch Mittel zum Verbinden von Signalen in der Anordnung mit den rekonfigurierbaren E/A-Anschlußtreibern 153, die die Hauptschnittstelle zu einer RSP-Vorrichtung darstellen. Jeder Anschluß am RSP weist eine programmierbare Schnittstellenschaltung auf, die durch ein Konfigurationsregister gesteuert wird. Damit kann jeder Anschluß eine Vorgabebedeutung (wie beispielsweise eine der Eingangsleitungen vom Schaltbus) oder eine rekonfigurierte Verwendung (Logikeingang, Logikausgang oder zweiseitig gerichteter Dreizustandsbustreiber) aufweisen. In der Figur sind rekonfigurierbare E/A-Anschlußmittel 153 als die Leitungen "A", "B", "C", "D", MPU, und "Q" gezeigt. Auch werden an der Peripherie der konfigurierbaren Logikanordnung detaillierte Steuersignale typischerweise zur Verfügung gestellt. In der zweiten logischen Schicht sind zur Implementierung der Pipeline Stufen einer typischen Anwendung konfigurierbare Takt- und Rücksetzsignale über die gesamte Anordnung verteilt.

Die dritte logische Schicht der konfigurierbaren Logikanordnung (ALP) implementiert die Pipeline-Busanordnung (PBA) 132 und Pipeline-Steueranschlüsse (PCP) 134. Die Signalleitungen in dieser Schicht überqueren das gesamte horizontale Ausmaß der Anordnung und teilen LS die Anordnung in zwei logische Bereiche ein. Die PBA-Leitungen durchqueren das "Datenweg"-Gebiet 156, während die PCP-Leitungen das "Steuer"-Gebiet 158 überqueren. Im allgemeinen Fall ist eine spezifische ALP-Schaltung ein "Pipeline Segment" 160 genanntes rechteckiges Gebiet der Anordnung, das einen Datenweg enthält, der mit einem oder mehreren PBA-Bussen und einem Steuerabschnitt, der an einen oder mehrere PCP-Leitungssätze angeschaltet ist, verbunden ist. Im allgemeinen können gleichzeitig mehrere Pipeline Segmente im ALP resident sein und jedes beliebige Segment kann rekonfiguriert werden, ohne die Operation der übrigen Segmente zu beeinflussen. Der Vorgang des Änderns von nur einem Teil der Logikanordnung ohne Beeinflussung der Operation des übrigen Teils wird "teilweise Rekonfiguration" genannt. Teilweise Rekonfiguration erlaubt Optimierung der Nutzung des konfigurierbaren Logikmittels durch die Entwicklungssoftware. Die Verbindung der Kernzellen- und Lokalbus signale mit den PBA- und PCP-Signalen wird durch horizontale Verstärkerschaltungen durchgeführt, die eine direkte Verbindung eines Lokalbussegments mit einer beliebigen der möglichen PBA-Bitleitungen erlauben, die die Zeile der Anordnung überqueren. Dadurch kann ein Pipeline Segment ohne ein Erfordernis zusätzlicher Signalwegführung auf jede beliebige horizontale Lage in der Anordnung verlegt werden. Durch dieses Wiederverlegungsmerkmal wird die automatische Erzeugung der zur Durchführung einer Berechnung benötigten konfigurierbaren Logikschaltungen dramatisch vereinfacht.

Jede Kernzelle 150 der ALP-Anordnung 120 kann zur Implementierung einer beliebig großen Anzahl von Logikfunktionen programmiert werden. Dazu gehören Operationen wie logisches UND, ODER, XOR, Multiplexer- und Flipflop-Operationen. Wie bemerkt, erlaubt die Peripherie der Kernzellenanordnung die Verbindung mit den programmierbaren E/A-Anschlüssen 153 an einem RSP-Baustein. Jede Kernzelle kann (mittels der programmierbaren Verbindungsschalter) zum Zugreifen auf eines oder zwei Signale aus den vier Nachbarkernzellen programmiert werden. Jede Kernzelle kann auch Daten auf einem (in der Figur mit gestrichelten Linien gezeigten) Lokalbusanordnungs-(LBA-)Segment empfangen oder übertragen. Wie bemerkt, bilden die LBA-Verbindungen für eine Teilmenge der Kernzellen einen (bereits erwähnten) sogenannten Kernblock. Der Kernzelle und dem LBA-Netz überlagert ist die Pipeline-Busanordnung (PBA) 132, die Signale führt, die die gesamte Anordnung in der horizontalen Richtung überqueren. Wie bemerkt, erlauben "Verstärker" genannte Schaltungen die Weiterführung der LBA-Daten zu Nachbarblöcken, den programmierbaren E/A-Anschlüssen und der PBA. Ein Satz programmierbarer Takt- und Rücksetzsignalgeneratoren 154 überqueren die Anordnung in der vertikalen Richtung. Diese Signale ermöglichen, daß jede Spalte als Pipeline Stufe in der Anordnung fungiert. Die PBA-Datenleitungen 132 erlauben Verbindung mit den Bussen RD, WD, RX, WX, RA und WA im RSP. Die PCP-Steuerleitungen 134 stellen programmierbare Steuer- und Statusleitungen dar, die zur ALP-Pipelinesteuerung direkt mit der RPIC verbunden sind. Die Fig. 5 zeigt den typischen Bereich, der von einem programmierbaren Pipeline Segment 160 einer Berechnung benutzt wird. Jedes Pipeline Segment besteht aus einer Anzahl von Pipeline-Datenwegstufen im "Pipelinedatenweg"-Bereich 156 des ALPs und den zugehörigen Pipeline Steuerschaltungen im "Pipeline Steuerbereich" 158.

Wie bemerkt, besteht ein einmaliges Merkmal des ALP darin, daß die Definitionen der LBA-PBA-Verbindung im Pipeline Segment Konfigurationsblock erlauben, daß sich das Pipeline Segment an einer beliebigen horizontalen Stelle im ALP befindet. Dadurch können gegenwärtig unbenutzten Pipeline Segmenten je nach Bedarf der aktuellen Aufgabe im Anwendungsprogramm neue Pipeline Segmente überlagert werden. In einem RSP-basierenden System wird die detaillierte Konfiguration und Programmierung der benötigten Pipeline Segmente während des Programmkompileierungsprozesses erstellt. Die zur Konfigurierung eines Pipeline Segments benötigten Daten werden als Konfigurationsblock (CB-Configuration Block) bezeichnet. CB-Daten werden im Programmadreßraum des RSPs gespeichert und unter Verwendung von Konfigurationssteuerbefehlen in den ALP eingeladen.

Die rekonfigurierbare Pipeline-Befehlssteuerung (RPIC Reconfigurable Pipeline Instruction Control) ist dafür verantwortlich, die Reihenfolge der Befehlsausführung unter Verwendung der Programmadressbusleitungen (PA) zu bewahren, auszuführende Befehle vom Speicher unter Verwendung der Leseprogramm-(RP-Read Program) Busleitungen zu empfangen und den Pipeline-Datenfluß in der Pipeline-Busanordnung zu koordinieren. Auch führt die RPIC vom Schaltbuskoppler empfangene Befehle zur Implementierung von Parallelverarbeitungsstrukturen aus. Wie bemerkt, bilden die übrigen sechs Busse (siehe Fig. 3) die pipeline-Busanordnung (PBA), die gleichzeitigen Datenfluß zwischen der Parallelverarbeitungsverbindungsstruktur (TBT), dem konfigurierbaren Logikmittel (ALP), dem Speichermittel (MPM) und dem skalaren Prozessor (PDP) erlaubt. Im allgemeinen Fall kann die PBA mehr oder weniger Busse als gezeigt aufweisen, sollte aber mindestens einen Datenadressbus, eine interne Datenbusverbindung mit dem skalaren Prozessor, eine Busverbindung zum Speicher und eine Verbindung zu der Parallelverarbeitungsverbindungsstruktur enthalten. In dem besprochenen Beispiel besteht die PBA aus sechs Bussen für Speicherleseadresse (RA-read address), Speicherschreibadresse (WA-write address), Schaltbusdateneingabe (RX), Schaltbusdatenausgabe (WX), LeseDaten (RD-read data) und Schreibdaten (WD-write data). Die Busse RD und WD sind jeweils in zwei getrennte ALP-Block-interne Busse aufgeteilt. Dies ermöglicht einen getrennten Datenfluß zwischen dem ALP-Block, dem Speicherblock und dem skalaren Prozessor. Im Fall der allgemeinen Architektur können der PBA zusätzliche Busse hinzugefügt werden, einschließlich beispielsweise der Programmadresse-(PA) und Leseprogramm-(RP-) Busse hinzugefügt werden.

Der Pipelinesteuerblock steht in Wechselwirkung mit dem skalaren Prozessor zur Ausführung von standardmäßigen Befehlen unter Verwendung von herkömmlichen Pipeline-Steuerkonzepten. Der RPIC-Block steht mit dem konfigurierbaren Logikmittel unter Verwendung eines als Pipeline-Steueranschlüsse (PCP-Pipeline Control Ports) bezeichneten Satzes von Leitungen in Wechselwirkung. Jeder PCP erlaubt, daß eine konfigurierbare Logikschaltung im ALP mit den außerhalb der konfigurierbaren Logik vorkommenden Pipeline-Datenströmen in Wechselwirkung tritt. Typischerweise besteht eine PCP-Menge von Steuersignalen für jeden Pipeline-Busanordnungsbuss. Die Grundoperationen jedes Pipeline-Steueranschlusses, die Implementierungen der RSP-Architektur gemeinsam sind, sind folgende:

Funktionsanforderung: Die RPIC erkennt Befehle im Programm, die im Gegensatz zu einer Ausführung im skalaren Prozessor PDP durch die Schaltungen im ALP auszuführen sind. Die RPIC zeigt diesen Zustand der ALP-Schaltung an, indem sie ein Signal im PCP ändert. Wenn die ALP-Schaltung für den Befehl nicht bereit ist, wird die RPIC die Befehlsausführungspipeline aufhalten. Man beachte, daß die RPIC NICHT auf die Vollendung einer Befehlsausführung im ALP wartet, da eine typische ALP-Funktion mehrere Taktzyklen erfordert. Die RPIC kann im Gegenteil mit jedem Taktzyklus eine neue Funktionsausführung einleiten. Dies ist der typische Fall für die Implementierung der tiefen Pipelines in einer Anwendung. Die Funktionsausführungssteuerfähigkeit wird auch zur Implementierung der Fähigkeiten zur Erweiterung des Befehlsvorrats für eine Anwendung benutzt.

Programmdatenanforderung: Diese Art von Übertragung wird dann durchgeführt, wenn ein skalarer Befehl auf einem ALP-Bus als die Adresse für Daten im Speicher oder als die Daten für einen Befehl Bezug nimmt. Diese werden als "konfigurierbare Adressmodi" bezeichnet. Beispielsweise kann die Ausgabe einer ALP-Pipeline direkt als Operand für einen Befehl "ADD" im Skalarprozessor benutzt werden. Als Alternative können Daten unter Verwendung dieses Mechanismus direkt vom skalaren Prozessor zu einer ALP-Schaltung übertragen werden. Abschließend kann der Befehl des skalaren Prozessors die Ausgabe einer ALP-Pipeline als Adresse für eine Speicherschreib- oder Leseoperation benutzen. Durch den RPZC-Block wird die Reihenfolge der Datenübertragung bewahrt, indem gegebenenfalls Wartezyklen eingespeist werden, um ordnungsgemäßen Pipeline-Betrieb aufrechtzuerhalten. Der Steuermechanismus der Programmdatenanforderung wird zur Implementierung von Fähigkeiten der Verbesserung des Befehlsvorrates und komplizierter Adreßerzeugung für eine Anwendung benutzt.

Selbständige Datenanforderung: Eine ALP-Schaltung kann Datenübertragungen zu/von dem Speichermittel und zu/von dem Schaltbuskoppler unabhängig von der Programmbefehlsfolge anfordern. Diese werden "selbstständige" Datenübertragungen genannt. In einem typischen Fall wird die gesamte Berechnung in einer Gruppe von RSP-Vorrichtungen unter Verwendung des Mechanismus der selbstständigen Datenübertragung ohne Befehlsausführung für die einzelnen Schritte durchgeführt. Zur Ausgabe von Daten zum Speicher oder dem Schaltbus aktiviert die ALP-Schaltung ein PCP-Signal zur RPZC, das eine Ausgabeübertragung anfordert. Der RPIC-Block unternimmt dann die notwendigen Schritte zum Einschreiben der Daten in den Speicher oder Ausgeben der Daten über den Schaltbus. Wenn das gewünschte Mittel "besetzt" ist, muß die ALP-Schaltung weitere Operationen solange aufhalten, bis das PCP-Signal anzeigt, daß das Mittel verfügbar ist. Eine ähnliche Wechselwirkung tritt für Speicherdatenlese- und Schaltbusdatenleseoperationen ein. Selbstständige Datenübertragungen werden zur Implementierung von Pipelineverkettung von einer RSP-Vorrichtung zu einer anderen über den Schaltbus benutzt. Der Mechanismus der selbstständigen Übertragung wird auch dazu benutzt, direkten Zugriff zum Speichermittel durch eine oder mehrere Stufen in einer tiefen Pipelinestruktur zu erlauben.

Im allgemeinen Fall enthält der Pipeline-Steueranschluß (PCP) typischerweise zusätzliche Signale. Bei der beispielhaften RSP-Implementierung, die besprochen wird, werden folgende zusätzliche Operationen unterstützt:

Pipeline-Zustand: Die konfigurierbare Logikschaltung erzeugt ein oder mehrere status-Zustandssignale, die durch den Befehlsausführungsstrom unter Verwendung von Befehlen "bedingter Sprung" überwacht werden können. Diese Fähigkeit wird als Teil des normalen Rechenvorgangs zur Auswahl von alternativen Befehlsflußströmen benutzt.

Pipeline-Unterbrechung: Die konfigurierbare Logikschaltung erzeugt eine Unterbrechungsanforderung an den

RPIC-Block, die die Ausführung eines spezifischen Softwareunterprogramms bewirkt. Die Unterbrechung wird durch die Ausführung eines Funktionsanforderungsbefehls an den PCP bestätigt, der die Unterbrechungsanforderung erzeugt. Mit diesem Mechanismus kann die ALP-Schaltung den skalaren Prozessor als Teilprozessor für die konfigurierbare Logik benutzen. Diese Fähigkeit wird auch zur Implementierung von eingabe-/ausgabevorrichtungshängigen Berechnungen benutzt.

Wie bemerkt, ist die rekonfigurierbare Pipelinebefehlssteuerungs-(RPIC-)Einheit im RSP verantwortlich für die Steuerung der Programmausführung, Koordinierung der Tätigkeiten des Datenflusses zwischen den Verarbeitungseinheiten und Konfigurierung der programmierbaren Schaltungen im ALP. Die Grundfunktionsoperationen der RPIC lassen sich in fünf Kategorien einteilen:

Einschaltinitialisierung — Setzen der Steuerlogik-Flipflops und programmierbaren Logik auf Vorgabezustände. Dieser Vorgang kann auch nach der Einschaltinitialisierung durch Aktivieren eines externen Konfigurationsfreigabesignals oder durch Ausführung des Befehls "REBOOT" eingeleitet werden.

Dynamische Rekonfigurierung — RSP-Befehle erlauben, daß alle oder ein Teil der ALP-Schaltungen während der Programmausführung geändert werden können.

Programmfluß — Auslesen der erforderlichen Befehlsfolge für das Anwendungsprogramm aus dem MPM. Dazu gehört auch die Ausführung von "Sprung-" und "Aufruf-"Befehlen, die den aktuellen Wert des Programm-Adreßregisters (PAR) ändern.

MPM-Datenzuariff — Erzeugt die Adressen für Datenlese- und schreiboperationen am MPM. Die konfigurierbaren Adreßmodi (CAM — Configurable Address Modes) ermöglichen der RPIC die Benutzung von durch ALP-Pipeline-segmente erzeugten Adressen.

Pipeline-Synchronisierung — Datenflüsse im ALP, MPM, PDP und TBT werden als unabhängige Pipelineketten behandelt. Die RPIC erzwingt den Pipeline-Verzögerungszustand durch Einfügen von Wartezuständen in jeder Situation, in der die Daten nicht in einem bestimmten Taktzyklus zur Verfügung stehen.

Für die folgende Besprechung wird angenommen, daß die Operation der Pipeline im PDP auf der in Fig. 6 dargestellten klassischen Befehlsausführungsfolge nach Von Neumann basiert. Solange kein Konflikt besteht, wird in dem "Abruf-"Zustand in jedem Taktzyklus ein neuer Befehl ausgelesen. Jeder Befehl schreitet dann durch den "Decodier-"Zustand im nächsten Taktzyklus fort. Bei normalen PDP-Befehlen (wie beispielsweise in einen Akkumulator addieren) werden die Daten beim dritten Taktzyklus des Befehls aus dem MPM ausgelesen und dann wird der Befehl schließlich beim vierten Taktzyklus ausgeführt. Vom Programm eingeleitete Befehlsausführung in einem ALP-Pipeline-segment bedient sich bis zu dem Punkt, an dem der Befehl ausgeführt wird, desselben Schemas. ALP-Pipeline-segmente erfordern im allgemeinen einen zusätzlichen Ausführungstaktzyklus für jede Pipelinestufe im Segment. Die RPIC wartet nicht, bis die Daten alle Stufen einer ALP-Pipeline durchlaufen haben. Wenn ein Befehl ausgeführt wird, der sich auf einen ALP-Pipelinebusgebrauch bezieht, wird die RPIC die Befehlsfolgenpipeline solange aufhalten, bis die Pipelinesegmentsteuerschaltung im ALP den Zustand "Pipeline bereit" anzeigt. RSP-Befehle, die Daten aus dem ALP aus lesen, und die konfigurierbaren Adreßmodi erfordern gleichermaßen die Signale "bereit" von der konfigurierbaren Pipelinesteuerschaltung, ehe sie mit der Operation fortschreiten.

Die RPIC ist auch für die Koordinierung der selbstständigen Pipeline-segment-(APS-Autonomous Pipeline Segment)Operationen verantwortlich. Jedes ALP-Pipeline-segment kann so ausgelegt werden, daß es unabhängig von der normalen Programmausführungsfolge arbeitet. Das heißt, ein APS erfordert keine Befehlsausführung, um durch die Pipelinefolge fortzuschreiten. Schaltungen des APS-Typs werden hauptsächlich für asynchrone Schnittstellen zu externen Sensoren und Betätigungsgliedern benutzt, aber sie werden auch für die Schaltbussteuerung des C-Vektors für das Bushauptsteuerungs-RSP-Bauelement in einer SIND-Anordnung benutzt. Die "Bereit" Signalschnittstelle wird dann von der RPIC zur Synchronisierung der Programmausführung mit dem APS-Betrieb benutzt.

Wie bemerkt, basiert die Steuerschnittstelle zwischen der RPIC und den Pipeline-segmenten im ALP auf einem Pipeline-Steueranschluß (PCP) genannten Protokoll. Für eine spezifische RSP-Implementierung können gleichzeitig mehrere PCP bis zu einer Höchstzahl im ALP aktiv sein. Die RPIC/ALP-Steuerschnittstelle besteht aus einer Menge von Pipeline-Busanordnungs-(PBA-Pipeline Bus Arrangement)Signalen (die die PCP-Signale bilden), die die Länge des Steuerabschnitts des ALP durchlaufen. Diese Steuerschnittstelle wird konzeptionell in der Fig. 7 dargestellt, die die Steuerschnittstelle zwischen den RPIC- und ALP-Modulen und die für jeden Pipeline-Steueranschluß (PCP-pipeline control port) benötigten (mit PCPj in der Figur markierten) Signale zeigt. Für die gegenwärtige Besprechung wird ein RSP-Bauteil mit höchstens acht PCP angenommen. Die Funktion jedes der Pipeline-Steueranschlüsse ist in Tabelle 2 zusammengefaßt.

Tabelle 2

Beispiel eines Satzes Pipelinefunktionen

Nnummer	PCP-Name	Anschlußfunktion
0	PSA	Allgemeines Pipelinesegment 0
1	PSB	Allgemeines Pipelinesegment 1
2	PSRA	Leseadresse-Pipelinesegment
3	PSWA	Schreibadresse-Pipelinesegment
4	PSRD	Lesedaten-Pipelinesegment
5	PSWD	Schreibdaten-Pipelinesegment
6	PSWX	Schreib-Pipelinesegment für externe Daten
7	PSRX	Les-Pipelinesegment für externe Daten

Die Steueranschlüsse PSRA, PSWA, PSRD und PSWD werden zur Implementierung der konfigurierbaren Adreßmodusoperationen benutzt. Die Steueranschlüsse PSWX und PSRX ermöglichen die Implementierung von selbstständigen Pipelines für Datenübertragungen über den Schaltbus. Die Steueranschlüsse PSA und PSB haben keine festgelegte Funktionsbedeutung. Es können mehr als ein Steueranschluß kombiniert werden, um einen logischen Steueranschluß mit getrennten Steuerungen für unterschiedliche Operationen innerhalb dieses Segments zu bilden. Beispielsweise kann die ALP-Schaltungsimplementierung die Steueranschlüsse PSRA und PSWA kombinieren, so daß dieselbe Schaltung zur Erzeugung von sowohl Schreib- als auch Leseadressen benutzt wird. Die RPZC liefert alle 16 Bit des Befehlsregisters an den ALP. Die ALP-Steuerschaltungen decodieren dann das gesamte Wort oder einen Teil desselben, um die Operation für einen bestimmten Befehl zu spezialisieren oder um die Anzahl von durch ein einziges Pipelinesegment durchgeführten Funktionen zu erweitern. Wie bemerkt, zeigt die Fig. 7 die sechs für jeden Pipeline-Steueranschluß (PCP-pipeline control port) benötigten Signale:

ACTj: Aktives Pipelinesegment — wenn durch die ALP-Steuerschaltung für diesen Pipelineanschluß auf LOW gesetzt, zeigt dieses Signal an, daß der Pipelineanschluß aktiv ist und auf die anderen RPIC-Anschlußsteuerungen reagieren wird. Wenn dieses Signal auf HI belassen wird, dann ist die Anschlußsteuerschaltung nicht verfügbar und die RPIC erzeugt eine Falle unzulässiger Befehl, wenn das Programm versucht, diesen Pipelineanschluß anzusteuern. Dies wird zur Fehlersuche in Anwendungsprogrammen benutzt.

FRQj: Funktionsanforderung (Function Request) an Anschluß j — ein Signal mit aktivem LOW zeigt der ALP-Steuerschaltung an, daß das IR- (Instruction Register) Wort gültig ist und daß die Steuerschaltung mit der Ausführung des Befehls beginnen sollte.

FRYj: Funktion bereit (Function ready) für Anschluß j — ein Signal mit aktivem LOW zeigt an, daß die ALP-Steuerschaltung den vom FRQj-Signal angeforderten Befehl genommen hat. Die RPIC wird keinen weiteren Befehl an den ALP ausgeben, bis dieses Signal auf LOW gesetzt ist.

DRQj: Datenanforderung (Data Request) für Anschluß j — ein Signal mit aktivem LOW zeigt an, daß die RPIC Daten für eine Schreibfunktion bereit hat oder Datenausgabe für eine Lesefunktion anfordert. Dies entspricht dem Zustand "Daten" in der normalen Pipelinebefehlsausführungsfolge.

DRYj: Daten bereit (Data Ready) für Anschluß j — ein Signal mit aktivem LOW zeigt an, daß die ALP-Steuerschaltung Daten zum Lesen bereit hat oder die Schreibdaten als Eingabe angenommen hat. Für Daten, die zum ALP übertragen werden, wird die RPIC eine nachfolgende Datenanforderung solange aufhalten, bis dieses Signal auf LOW gesetzt ist. Für Daten, die vom ALP übertragen werden, hält die RPIC die Befehlspipeline solange auf, bis dieses Signal gesetzt ist und eine gültige ALP-Ausgabe anzeigt.

ARQj: Selbständige Anforderung (Autonomous Request) j — ein Signal mit aktivem LOW zeigt an, daß die ALP-Steuerschaltung eine selbstständige Datenübertragung auf dem zugehörigen Datenbus anfordert.

ARYj: Selbstständig bereit (Autonomous Ready) für Anschluß j — ein Signal mit aktivem LOW von der RPIC zur Pipeline-Steuerschaltung; zeigt an, daß die angeforderte Übertragung abgeschlossen ist.

STATj: Status für Anschluß j — die RPIC benutzt dieses Signal zur Implementierung von bedingten Sprungbefehlen, die von dem aktuellen Zustand des Pipeline-segments abhängig sind. Die Bedeutung des Zustandssignales ist abhängig von der ALP-Schaltung und entspricht einem nützlichen Zustand in der Anwendung.

Beispielsweise kann das STAT-Signal anzeigen, daß externe Daten zur Verfügung stehen, einen Arithmetik-Überlauf usw. Das STAT-Signal für einen Steueranschluß kann durch die Konfigurierungsfolge zur Implementierung einer asynchronen Unterbrechung zur RPIC programmiert werden.

Für allgemeine Pipelinesegmente im ALP erfordert die normale konfigurierbare ALP-Befehlsausführung die Verwendung von sowohl den FRQ/FRY, als auch den DRQ/DRY-Steuerungen. Daten zu und von für konfigurierbare Adreßmodi benutzten Pipelines erfordern nur die DRQ/DRY-Steuersignale. Selbstständige Pipelines

benutzen die ARQ/ARY-Signale.

Die Fig. 8 zeigt ein Anwendungs-Pipeline-segment, das eine Addier-/Akkumulierfunktion durchführt. Diese Pipeline weist eine einzelne Stufe auf, so daß die Pipelinefunktion in einem einzelnen Taktzyklus durchgeführt wird. Das Segment benutzt zwei PCP, einen für Dateneingabe und einen für Datenausgabe. Die Leitungen ACT0 und ACT1 werden auf LOW gesetzt, um anzuzeigen, daß die PCP aktiv sind. Die Funktion ist stets für beide Anschlüsse bereit, so daß die Leitungen FRY0 und FRY1 ebenfalls auf LOW gesetzt werden. Gleichmaßen steht das Pipeline-segment stets zum Addieren eines neuen Wertes und zur Ausgabe eines Wertes zur Verfügung, so daß die Signale DRY0 und DRY1 ebenfalls auf LOW gesetzt werden. Jedesmal wenn ein Befehl "laden ALP" ausgeführt wird, wird das DRQO-Signal abgetastet, um anzuzeigen, daß neue Daten verfügbar sind. Dieses Signal wird dann zur Freigabe des Pipelinetaktes für die Pipeline-Registerspalte benutzt. Die RPIC-Einheit enthält Konfigurationsregister, die die Fähigkeit jedes pipeline-Steueranschlusses anzeigen. Die Steuerbit für jeden PCP sind aus Tabelle 3 ersichtlich:

Tabelle 3

RPIC-Steuerregisterbit für PCP-Steuerung

PCP-Steuerbit	Bedeutung
j+0	Aktive Pipelinesteuerung für diesen Anschluß; dies ist eine redundante Anzeige, daß für diesen Anschluß eine Pipelinesteuerschaltung besteht. Die RPIC erfordert, daß dieses Bit gesetzt ist, sowie das ACT-Signal zur Freigabe der Pipelineaktion.
j+1	Unterbrechungsfreigabe - Das Statussignal für diesen Anschluß ist ein Unterbrechungssignal mit aktivem LOW. Dieses Signal veranlaßt, daß die RPIC die aktuelle Programmausführung anhält und ein mit der Unterbrechungsnummer für diesen Anschluß verbundenes Unterbrechungsdienstprogramm ausführt. Die Pipelinesteuerschaltung muß die Unterbrechungsanforderung als Reaktion auf eine Befehlsausführung unter Verwendung der Steuerungssignale FRQ/FRY löschen.
j+2	Typ des Pipeline-segments: 0 zeigt ein befehlsynchronisiertes Pipeline-segment an. 1 zeigt an, daß dies ein selbstständiges Pipeline-segment ist.
j+3	Gültig-Zustand - zeigt an, daß die PCP-Statusleitung als gültiger Statuszustand für bedingte Befehle angesteuert wird.

Die Operation von selbstständigen Pipelines ist von dem PBA-Datenbus abhängig, der von der durch die Anschlußnummer definierten Pipeline benutzt wird. Ein selbstständiger Anschluß kann mit anderen Anschlüssen kombiniert werden, um eine komplexe Pipeline-segmentsteuerschnittstelle zu erstellen.

Die Tabelle 4 zeigt eine beispielhafte Menge von selbstständigen Übertragungen für jeden Steueranschluß. Die selbstständigen Pipelines PSWX und PSRX können zur Verkettung von Pipelines von einem RSP-Bauelement zu einem anderen benutzt werden. Die selbstständigen Pipelines PSRD und PSWD werden normalerweise für Preprozessor- und Postprozessor-Datenhandhabung für Berechnungen unter Befehlssteuerung im PDP benutzt. Die Operationen werden unter Verwendung der durch Befehl eingeleiteten Steuerungssignale DRQ und DRY für die zugehörige Pipeline synchronisiert. Die selbstständigen Übertragungen PSA und PSB werden typischerweise für Sensordateneingabe und Betätigungsgliedsteuerung benutzt, die hochratige Datenübertragungen erfordern, die im Bezug auf die aktuelle Programmausführungsaufgabe als "Hintergrund-"Aktivität durchgeführt werden können. Beispielsweise wird ein Datenblock in den MPM eingelesen, während ein vorheriger Block durch das Programm verarbeitet wird. Die Konzepte "verzweigen" und "verbinden" der parallelen Verarbeitung werden zur Synchronisierung der selbstständigen und Programmausführungsaufgaben benutzt. Die selbstständigen Pipelines PSRA und PSWA werden zur Implementierung des Äquivalents von "Speicherdirektzugriffs-"(DMA — direct memory access) Übertragungen über den Schaltbus benutzt.

Tabelle 4

Selbstständige Pipeline-Datenübertragungen

Nummer	PCP-Name	Selbstständige Übertragung
0	PSA	Allgemeines Pipelinesegment 0; E/A-Schreib-Pipelineaktion vom MPM zu rekonfigurierbaren Eingabe-/Ausgabeleitungen (wie z.B. der A- oder B-Bus). Der RA-Bus definiert die Adresse für die Daten. ARY wird gesetzt, wenn die Daten verfügbar sind und ARQ wird fallengelassen, wenn die Daten aufgenommen sind.
1	PSB	Allgemeines Pipelinesegment 1; E/A-Lese-Pipelineaktion von einem rekonfigurierbaren Eingabe-/Ausgabebus (wie beispielsweise dem A- oder B-Bus) zum MPM. Der WA-Bus definiert die Adresse zum Speichern der Daten. ARY wird gesetzt, wenn die Daten geschrieben worden sind.
2	PSRA	Leseadref-Pipelinesegment; ARQ für diesen Anschluß leitet eine Datenübertragung vom MPM zum Schaltbus ein. ARY wird LO gesetzt, wenn die Übertragung abgeschlossen ist.
3	PSWA	Schreibadressen-Pipelinesegment; ARQ für diesen Anschluß leitet eine Datenübertragung vom Schaltbus zur MPM-Schreibadrefsstelle ein. ARY wird LO gesetzt, wenn die Übertragung abgeschlossen ist.
4	PSRD	Lesedaten-Pipelinesegment; der RA-Adrefbus liefert die Adresse für eine MPM-Leseoperation. ARY wird LO gesetzt, wenn die Daten zum ALP verfügbar sind. Die ALP-Steuerschaltung muß das ARQ-Anforderungssignal freigeben, wenn die Daten aufgenommen sind.
5	PSWD	Schreibedaten-Pipelinesegment; der WA-Adrefbus liefert die Adresse für eine MPM-Schreiboperation. ARY wird auf LO gesetzt, wenn die Daten vom ALP aufgenommen worden sind. Die ALP-Steuerschaltung muß das ARQ-Anforderungssignal freigeben, wenn die Daten aufgenommen sind.
6	PSWX	Übertragung von Daten zur Schaltbuseinheit; ARY zeigt an, daß die Daten vom Schaltbus aufgenommen sind.
7	PSRX	Lesen externer Daten vom Schaltbus zum Pipelinesegment. ARY wird auf LO gesetzt, wenn Daten vom TBT zum ALP verfügbar sind. Die ALP-Steuerschaltung muß das ARQ-Anforderungssignal freigeben, wenn die Daten aufgenommen sind.

Die Schaltbussteuerungen für die selbstständigen Übertragungen, die den TBT adressieren, werden durch Register in der RPIC-Einheit definiert. Die Werte dieser Register werden durch normale Befehlsausführung definiert. Die RPIC führt eine Befehlsmenge aus, um die, ALP-Pipeline-Verarbeitungsfähigkeit mit den Arithme-

ticverarbeitungsfähigkeiten des PDP und den Eingabe-/Ausgabeübertragungen zu synchronisieren. Für diesen Zweck werden acht Hauptbefehle benutzt, einer für jeden Pipeline-Steueranschluß. Von der ALP-Steuerschaltung können zusätzliche Befehlsregister-(IR-)Bit decodiert werden, um eine große Menge von anwendungsabhängigen Befehlen zu erstellen. Die direkten Befehle für diesen Zweck sind in der Tabelle 5 für die Beispielmengen von Pipeline-Datenwegen dargestellt.

Datenübertragung zwischen dem PDP und ALP werden stets unter Verwendung der konfigurierbaren Adreßmodenfähigkeit synchronisiert. Der PFS-Befehl erlaubt einer zusammenhängenden Menge von Pipeline-Flipflopwerten in eine Speicherstelle eingelesen zu werden. Die Lese-Zeilen-/Spaltenadresse in der Anordnung ist im RPIC-Konfigurationsadreßregister enthalten, das durch einen Konfigurationssteuerbefehl besetzt werden kann.

Tabelle 5

Beispielhafte Pipeline-Steuerbefehle

Operati- onscode	Parameter	Befehlsfunktion
PCj	4-Bit-Funktion	Pipeline-Steuerung für Anschluß j; die Pipeline-Steuerschaltung leitet eine von sechzehn, als Reaktion auf den Befehl durchzuführenden Funktionen ein.
PPS	MPM-Adresse	Einlesen ALP-Pipeline-Registerwert in Speicher
PRA	MPM-Adresse	Übertragen Daten von MPM zum ALP mit RD-Bus, RA-Segment
PWA	MPM-Adresse	Übertragen Daten von ALP zum MPM mit WD-Bus, WA-Segment
PRD	MPM-Adresse	Übertragen Daten von MPM zum ALP mit RD-Bus
PWD	MPM-Adresse	Übertragen Daten von ALP zum MPM mit WD-Bus
PWX	-	Übertragen ALP-Daten zum TBT mit WX-Bus
PRX	-	Übertragen TBT-Daten zum ALP mit RX-Bus

Detailkonzepte der ALP-Anordnung

Die Kernzellenanordnung des ALP ist zwar symmetrisch, aber die Pipeline-Busstruktur ist zur Implementierung von mehrstufigen Pipeline-Datenwegen optimiert. Vertikale Takt- und Rücksetzleitungen ermöglichen Steuerung von Pipelinestufen mit verringerter Verlustleistung unter Verwendung eines Taktfreigabeschemas (eine Spalten-Takt- oder Rücksetzleitung wird nur dann abgefragt, wenn das Spalten-Taktfreigabe- oder Rücksetzfreigabesignal aktiv ist). Eine Auslegung der Kernzellen und der Gesamtanordnung, die zur Verwendung in der RSP-Architektur geeignet ist, ist die schon erwähnte CLAY-Familie konfigurierbarer Logikanordnungsbau-
steine, die von National Semiconductor Corporation in Santa Clara, Kalifornien, erzeugt wird. Für die Zwecke der vorliegenden Erfindung können auch andere konfigurierbare Logikmittel mit der erforderlichen Funktionalität (bzw. mit der Fähigkeit, dafür modifiziert zu werden) benutzt werden.

In der RSP-Architektur werden die globalen Takt und getasteten Taktmodi der CLAY-Anordnung bewahrt, so daß der ALP zur Steuerung von externen Sensor-/Betätigungsverrichtungen benutzt werden kann. Das Schema

mit freigegebenem Takt und Rücksetzen ist der bevorzugte Ansatz für Anwendungen mit rekonfigurierbarer Pipeline. Die PBA-Leitungen laufen rechtwinklig zu den Taktleitungen und ermöglichen direkten Zugriff auf die Pipeline-Busmittel der internen RSP-Speicher (MPM), des Schaltbuskopplers (TBT) und des Pipeline-Datenprozessors (PDP).

In der Fig. 9 ist die detaillierte Schnittstelle für eine ALP-Kernzelle 200 dargestellt. Im normalen Betrieb liefern die Leitungen "A" und "B" den Signalfluß des nächsten Nachbarn zwischen den Kernzellen. Jede Kernzelle 200 kann auf ihren (in der Figur mit Ln, Le, Ls und Lw bezeichneten) vier Seiten mit einem oder zwei lokalen Bussegmenten verbunden sein. Konfigurationsdaten für eine Kernzelle werden aus den vier Konfigurationsdatenleitungen auf der "Nord"-Seite (den Cdn-Leitungen) und den vier Konfigurationsleitungen auf der "Süd"-Seite (den CDs-Leitungen) ausgelesen. Dies ergibt acht Bit von Konfigurationsdaten zu einer Kernzelle in einem Schritt. Jede Kernzelle wird durch UND-Verknüpfung der vertikalen CSL-(column select low) Leitung und der horizontalen RS-(row select) Leitung als x/y-Stelle der Anordnung zum Laden des niedrigwertigen Bytes der Konfiguration für die Zelle adressiert. Das hochwertige Byte der Konfiguration wird durch die Kombination der CSE-(column select high) Leitung und der RS-Leitung gesetzt. Jede Spalte von Kernzellen bedient sich einer gemeinsamen Takt-(CLK-) und Rücksetz-(RST-)Leitung, die, wie im nächsten Abschnitt beschrieben wird, durch Schaltungen oben und unten an der Anordnung gesteuert wird.

Fig. 10 zeigt eine beispielhafte Schaltung zur Implementierung einer ALP-Kernzelle. Diese Schaltung ist funktionsmäßig der CLAY-Kernzelle gleichwertig. Die Schaltungslogik besteht aus zwei Abschnitten: (1) einer Konfigurationsdatenschaltung 202, die die Zellenfunktion und Eingangsverbindungen definiert und (2) einer Funktionsschaltung 201, die die durch die Konfigurationsdaten definierte Operation durchführt.

Die Konfigurationslogik für eine Kernzelle besteht aus einem 16-Bit-Register (D-Speicher 203 und F-Speicher 204) und einer Konfigurationsdecoderschaltung (DC) 206. Die Ausgabe des Konfigurationsregisters und Decoders werden zur Auswahl der Funktionssignalwege für die durch die Kernzelle durchzuführende Funktion benutzt. Beispielsweise werden Konfigurationsregister-bit R0 und R1 zur Auswahl der Logikeingabe "A" in die Zelle aus einer der vier Nachbarzellen benutzt. Die niedrigstwertigen acht Bit des Konfigurationsregisters werden dadurch geladen, daß zuerst die CSL- und SET-Signale gesetzt werden. Danach wird das Zeilenauswahlsignal (RS) abgetastet, um die Datenbit zu laden. Die höchstwertigen acht Bit werden auf ähnliche Weise unter Verwendung des CSE-Signals anstatt des CSL-Signals geladen. Der aktuelle Konfigurationsdatenwert läßt sich durch Setzen der Signale CSE bzw. CSL auf LOW von den CD-Leitungen ablesen. Der aktuelle Wert des Kernzellen-Flipflops läßt sich durch Setzen des Signals FRD lesen. Damit kann das Anwendungsprogramm einen Pipeline-Registerwert direkt lesen.

Freigegebene Takt und Rücksetzleitungen

Die Implementierung eines Takt- und Rücksetzfreigabeschemas vereinfacht den Aufbau von Pipelineketten, verringert die Größe jeder Pipelinestufe und verringert die Stromaufnahme, da ein Spaltentakt nur dann abgetastet wird, wenn das Freigabesignal gesetzt ist (nominell ein Wert mit Aktiv-HI). Dadurch kann eine globale Pipeline-Verzögerungsfähigkeit implementiert werden. Ein PIPEN genanntes vom ALP gesteuertes Signal kann STALL setzen, um Taktung aller Spalten, die Takt- und/oder Rücksetzfreigabe-gesteuert sind, zu sperren. Das Signal PIPEN hat keine Wirkung auf die Spalten, die globale Takt/Rücksetz- oder getastete Takt/Rücksetzsteuersignale benutzen.

Implementierung des Takt/Rücksetzfreigabeschemas erfordert Erweiterung der Takt/Rücksetz-Spaltensteuersignale an der oberen und unteren Peripherie der Anordnung. Der Unterschied besteht darin, daß jedes Flipflopelement einer durch Takt freigegebenen Spalte nur dann geladen wird, wenn das Spaltentaktfreigabesignal von der ALP-Steuerschaltung gesetzt und das PIPEN-Signal gesetzt ist. Alle Flipflopelemente (Flipflopkonfigurationen einer Kernzelle) werden auf dieselbe Weise beeinflußt. Das Spaltenrücksetzen kann ähnlicherweise durch ein vom ALP gesteuertes Rücksetzfreigabesignal freigegeben werden.

Die Fig. 11 zeigt eine beispielhafte Schaltung für die Spaltentaktleitung einer ALP-Zelle. Die Konfigurationssignale CC0, CC1 und CC2 steuern die Spaltentaktwirkung. Wenn CC0 = 1 und CC1 = 0, dann ist der Spaltentakt gesperrt. Wenn CC0 = 1 und CC1 = 1, dann ist der Spaltentakt die Ausgabe "A" der oberen Zelle der Spalte (die Konfiguration mit getastetem Takt). Wenn CC0 = 0 und CC1 = 1, dann wird der Spaltentakt vom südlichen Lokalbussegment der oberen Zeile angesteuert. Wenn CC0 = 0 und CC1 = 0, dann ist der Takt vom globalen Taktsignal abgeleitet und wird durch das Konfigurationssteuersignal CC2 und das Steuersignal PIPEN gesteuert. Wenn CC2 = 0, dann wird der Spaltentakt mit jedem Taktzyklus vom globalen Takt abgetastet. Wenn CC2 = 1, dann wird der Spaltentakt nur dann durch den globalen Takt abgetastet, wenn die Ausgabe "A" der oberen Zelle gesetzt ist. Dies ist der Fall der Pipeline-Taktfreigabe. Der Pipelinetakt kann global durch das Signal PIPEN gesperrt werden, wodurch alle Pipelines in der Anordnung verlangsamt werden.

Pipeline-Busanordnung

Mit der Pipeline-Busanordnung (PBA) wird eine Pipeline-Datenwegauslegung zum Zugreifen auf die Speicheradreß- und Datenleitungen des RSP ohne Erfordernis einer Wegführung von Signalen zur Peripherie der Anordnung ermöglicht. Dies ist ein Schlüsselmerkmal des RSPs zur wirkungsvollen Verwendung des ALP. Das PBA-Konzept basiert auf zwei Beobachtungen der Verwendung von CLAY-Vorrichtungen, die den meisten SRAM-basierenden FPGA gemeinsam ist: (1) es gibt ein ausgedehntes Verdrahtungsnetz (8 Drähte pro Spalte), das für Konfigurationsdatenverteilung benutzt wird und das nicht benutzt wird, wenn die Anordnung eine Anwendung fährt; und (2) die MODGEN-Funktionsmenge zeigte an, daß alle interessierenden Datenflußoperatoren wirtschaftlich unter Verwendung einer Datenflußstruktur mit "Zweiraster" implementiert werden können.

ten. Die Verwendung des PBA erlaubt die Entwicklung von dynamischen Pipeline-Grundelementen, die ohne Notwendigkeit einer zusätzlichen Wegeführung an jede Blockstelle im ALP geladen werden können. Mehrere durch MODGEN erstellte Pipelineblöcke können in unterschiedlicher Reihenfolge in den ALP geladen werden und führen nützliche Funktionen im System mit durch die PBA-Verbindungen vorgesehener "automatischer globaler Wegeführung" unter Verwendung von anstoßenden Nachbarverbindungen durch. Dadurch können auf Grundlage einer Basisbibliothek erzeugter Funktionen Tausende von Pipelinestrukturen implementiert werden. Rekonfigurierbare Logik, die nicht an der rekonfigurierbaren RSP-Pipelinestruktur teilnimmt, benutzt die PBA nicht und die PBA behindert auf keine Weise die Verwendung des ALPs für diesen Zweck.

Die Implementierung der PBA erfordert mehrere Verbesserungen der CLAY-Struktur (bzw. die Fähigkeit, die jeweils benutzte SRAM-basierende FPGA-Kernzellenarchitektur modifizieren zu können, damit sie wie beschrieben funktioniert), beeinflusst aber nicht die Auslegung der Kernzelle:

- Die Konfigurationsdatenwege müssen rechtwinklig zum Takt-/Rücksetzleitungsnetz liegen. Das erfordert die Verwendung eines Herstellungsprozesses mit drei Metallschichten.

- Die horizontalen Verstärker-(ER-horizontal repeater)Schaltungen werden erweitert, um Lesen und Ansteuern einer beliebigen der acht PBA-Leitungen für jede Zeile in der Anordnung zu ermöglichen. Vier von den PBA-Leitungen pro Zeile sind Pipeline-Busbit (DR, DW und so weiter). Die anderen vier PBA-Leitungen pro Zeile haben keine fest zugeordnete Bedeutung in der RSP-Blockschnittstelle und werden als globale Busse im ALP benutzt. Die ansteuerbaren Leitungen pro Zeile können durch Tristate-Treiber angesteuert werden, um eine zweiseitig gerichtete Busstruktur zu implementieren. Man beachte, daß PBA-Mittel nur in der horizontalen Richtung zur Verfügung stehen, so daß die Implementierung des PBA-Konzepts nur die horizontalen Verstärkerschaltungen beeinflusst.

- Um die Abtrennung von PBA-Treibern während einer Konfigurationsfolge zu ermöglichen, ist eine globale CFG-Leitung über die gesamte horizontale Verstärkeranordnung verteilt.

- Der linken und/oder rechten Seite der Anordnung sind Schaltungen hinzugefügt, um das Umschalten der Leitungen GBA vom Konfigurationsdatenverteilungsmodus in die Pipelinedatenwegeinstellungen zu erlauben. Zusätzliche Schalter werden dazu benutzt, daß ALP-Ausgangsdaten als eine "Phantom-E/A"-Datenquelle für PDP-Befehle benutzt werden. Mit dieser Einrichtung kann jeder PDP-Kernbefehl auf einen Ausgang des ALPs als Adressen- oder Datenelement als Teil der normalen Befehlsausführung zugreifen. Diese Fähigkeiten werden im Schema der RSP-Architektur als "konfigurierbare Adreßmodi", (CAM) bezeichnet.

Die Tabelle 6 zeigt eine Beispielliste des PBA-Gebrauchs im ALP. Man beachte, daß ungeradzahlige und geradzahlige Zeilen zur Unterstützung des Zweierasterkonzepts unterschiedlichen PBA-Gebrauch aufweisen (aufeinanderfolgende Wortbit sind durch eine Zwischenkernzelle getrennt). Man beachte, daß die Fig. 8 einen dynamisch verschiebbaren Addier/Akkumulier-Pipelineblock mit Verwendung der PBA als Datenquelle (DR-Bus) und eines zweiten PBA-Busses (DW) für die Pipeline-Ausgabe darstellt. Dieser Block (und Hundert tausende wie dieser) können ohne Notwendigkeit zusätzlicher Verdrahtungsmittel an jede horizontale Stelle des ALP gelegt werden. Dadurch vereinfacht sich das Problem der Platzierung und Wegeführung für rekonfigurierbare Rechensysteme. Der allgemeine verschiebbare Pipelineblock, der ein "Pipeline-segment" genannt wird, enthält die Steuerschaltungen, die mit dem Datenflußoperator verbunden sind. Die Segment-Steuerschaltung steht wie schon beschrieben mit der RPIC-Einheit in Wechselwirkung. Für den Zweck der vorliegenden Besprechung wird angenommen, daß die Steuerschaltungen von sich aus mit dem Datenfluß gebunden sind. Dies wird durch Verwendung von zusätzlichen PBA-Mitteln im Steuerteil des ALPs für Befehlsdecodierung, Statusantwort und Inter-Block-Steuerkommunikation erreicht. Zusammen genommen bilden diese Signale die Menge von Pipeline-Steueranschlüssen (PCP-Pipeline Control Ports).

Tabelle 6

Beispielhafte PBA-Belegung, ALP-Datenwegabschnitt

PBA-Leitung	Gerade Zeile	Ungerade Zeile	Bemerkungen
n0	R-F	R-F	Lesen Pipeline-Flip-Flops mit FRD-Merkmal
n1	R-RD	W-RD	Lesen und Schreiben am RD-Bus
n2	R-RA	R-WA	Leseadress- und Schreibadresswerte
n3	R	W	Nur ALP-Datenfluß
s0	R-WD	W-WD	Lesen und Schreiben am WD-Bus
s1	R	W	Nur ALP-Datenfluß
s2	R-RX	W-WX	Externe RX- und WX-Datenbusse
s3	R	W	Nur ALP-Datenfluß

Die Verteilung der Befehlsregister-(IR-instruction register)Daten und Pipeline-Steueranschluß-(PCP-)Signale im Steuerbereich des ALP basiert, wie teilweise in der Tabelle 7 gezeigt, auf einem anderen Ansatz.

Tabelle 7

Beispielhafte PBA-Belegung, ALP-Steuerabschnitt

5

10

15

20

25

30

35

ZEILE	PBA-Leitung	Signal	Bemerkungen
j	n0	R-IR0	Befehlsregister-Bit0
	n1	W-FRQ0	Funktionsanforderung PCP0
	n2	W-FRQ1	Funktionsanforderung PCP1
	n3	R/W	Nur ALP-Datenfluß
	s0	R/W	Nur ALP-Datenfluß
	s1	W-FRQ2	Funktionsanforderung PCP2
	s2	W-FRQ3	Funktionsanforderung PCP3
	s3	R/W	Nur ALP-Datenfluß
j+1	n0	R-IR1	Befehlsregister-Bit1
	n1	W-FRQ4	Funktionsanforderung PCP4
	n2	W-FRQ5	Funktionsanforderung PCP5
	n3	R/W	Nur ALP-Datenfluß
	s0	R/W	Nur ALP-Datenfluß
	s1	W-FRQ6	Funktionsanforderung PCP6
	s2	W-FRQ7	Funktionsanforderung PCP7
	s3	R/W	Nur ALP-Datenfluß

In dieser Anordnung sind die FRQ-Signale für alle PCP in zwei aufeinanderfolgenden Zeilen enthalten. Die FRY-Signale für alle PCP sind in den nächsten zwei Zeilen enthalten und so weiter. Dieses Muster wiederholt sich für alle PCP-Steuer-Signale und die Befehlsregister-Bit. Der gesamte Steuerbereich des ALPs erfordert 16 Zeilen für acht PCP-Mengen. Die PBA-Leitungen, die nicht im Datenflußbereich oder Steuerflußbereich liegen, können der Verwendung durch andere RSP-Steuer- und Datensignale zugewiesen werden.

45

Kernblockstruktur

Wie anhand der Fig. 5 besprochen, wird eine zusammenhängende rechtwinklige Anordnung von Kernzellen (CC) mit denselben lokalen Busverbindungen als ein Kernblock bezeichnet. Fig. 12 zeigt den allgemeinen Plan der Kernblockstruktur für einen Kernblock mit vier horizontalen Spalten und vier vertikalen Zeilen. Jeder Kernblock 220 ist unter Verwendung von horizontalen Verstärker-(HR-horizontal repeater) Schaltungen 222 und vertikalen Verstärker (VR-vertical repeater)-Schaltungen 224 mit Nachbarkernblöcken und mit der Pipeline-Busanordnung verbunden. Die Verteilung von Konfigurationsdaten wird durch horizontale Verstärkerschaltungen bewirkt, die die auf die gesamte Anordnung durchlaufenden PBA-Leitungen basierenden CD-Daten lesen oder schreiben. Bei einer kleinen ALP-Schaltung können die PBA-Leitungen direkt für die Konfigurationsdaten benutzt werden. Fig. 13 zeigt die Verbindungen, die einen Teil der Schnittstelle zwischen den Kernzellen und Verstärkerschaltungen an den Ecken der vier Kernblöcke bilden.

Fig. 14 zeigt eine beispielhafte Schaltung für eine vertikale Verstärkerschaltung 224. Jeder Verstärker empfängt Konfigurationsdaten von den horizontalen Verstärkerblöcken auf den CD-Leitungen. Jeder vertikale Verstärker erlaubt eine beliebige Verbindungskombination des lokalen Bussegments oberhalb des Verstärkers (La), des lokalen Bussegments unterhalb des Verstärkers (Lb), des Express-Bussegments oberhalb des Verstärkers (Ea) und des Express-Bussegments unterhalb des Verstärkers (Eb). Jede Spalte im ALP hat vertikale Verstärker links und rechts von der Spalte an jeder Kernblockgrenze.

Fig. 15 zeigt eine beispielhafte Schaltung einer horizontalen Verstärkerschaltung 222. Jede Zeile im ALP weist zwei horizontale Verstärkerschaltungen an einer Kernblockgrenze auf, eine für Bussegmente an der Nordseite und eine für die Bussegmente an der Südseite. Zusätzlich zu den Schaltungen, die die Verbindung der östlichen und westlichen lokalen und Express-Bussegmente (Le, Lw, Ee und Ew) erlauben, enthält der horizontale Verstärker Schaltungen zum Lesen oder Schreiben eines beliebigen der vier Pipelinebus-(PB-)Signale zum lokalen Bussegment an der Ostseite (Le). Die horizontalen Verstärker haben auch die Fähigkeit, die Konfigurationsdatenleitun-

gen(CD) vom PB anzusteuern oder die Konfigurationsdatenleitungen zurück zum PB zu lesen.

Detaillierter Daten- und Steuerfluß im RSP

Ein RSP-Programm greift auf die Parallelverarbeitungs-Zusammenschaltungen- (TBT-) und konfigurierbaren Logikanordnungsmittel (ALP) durch Erweiterungen des Befehlsvorrats des skalaren Prozessors (PDP) zu. Die Definition des Befehlsvorratzzugriffs auf Bitebene hängt von dem spezifischen benutzten skalaren Verarbeitungskern ab. Da alle skalaren Prozessoren eine auf der Architektur nach von Neumann basierende generische Befehlsvorratsform aufweisen, gilt die gegenwärtige Besprechung im allgemeinen für jeden skalaren Prozessor. Die Aspekte eines Befehlsvorratzzugriffs fallen in folgende Kategorien: (1) konfigurierbare Adreßmodi, (2) Funktionsausführungsbefehle, (3) bedingte Sprung-befehle, (4) konfigurierbare Logikunterbrechungen, (5) direkte Ein-/Ausgabe und (6) Rekonfigurationsbefehle.

Eine typische RSP-Implementierung erlaubt Zugriff auf Pipeline-Busanordnungsverbindungen unter Verwendung einer Erweiterung des integrierten indirekten Adressierungsmechanismus. Diese Erweiterungen der normalen Adressierungsfähigkeiten sind die konfigurierbaren Adreßmodi. Jeder Befehl im skalaren Prozessor, der einen Speicheroperanden ansteuert, kann die konfigurierbare Adreßmodusfähigkeit benutzen. Im RSP-Beispiel gibt es sechs mögliche konfigurierbare Adreßansteuerungen:

PAR-Lesen Pipeline-Schaltbuseingabe (Pipelined Toggle Bus Input Read). Dieser Adreßmodus erlaubt die direkte Ansteuerung von Schaltbuseingabedaten als Datenoperand.
PAW-Schreiben pipeline-Schaltbusausgabe. Dieser Adreßmodus wird mit Befehlen benutzt, die normalerweise zum Speicher schreiben. Die Datenausgabe wird als Ausgabewort vom RPIC-Block zum Schaltbus umgeleitet.
PAR-Lesen Pipeline-Adresse. Der durch die Ausgabe der Leseadrese-(RA-Read-Address) Pipelinebusanordnungsleitungen definierte Adreßwert wird als die Speicheradresse zum Lesen eines Operanden benutzt.
PAW-Schreiben Pipeline-Adresse. Der durch die Ausgabe der Schreibadressen-(WA-Write Address) Pipelinebusanordnungsleitungen definierte Adreßwert wird als Adresse einer Speicherstelle zum Schreiben von Daten definiert.

PDR-Lesen Pipeline Daten. Die Ausgabe der Lesedaten-(RD-Read Data) Pipelinebusanordnungsleitungen wird als Eingangsdatenoperand für den Befehl benutzt. Dies erlaubt die direkte Verbindung des konfigurierbaren Logikausgangs mit dem Eingang des skalaren Prozessors.

PDW-Schreiben Pipeline Daten. Die Ausgabe des skalaren Prozessors wird als Eingabe zu den Schreibdaten-(WD-Write Data) Pipelinebusanordnungsleitungen umgeleitet. Dies erlaubt die direkte Übertragung von Daten vom skalaren Prozessor zur konfigurierbaren Logik.

In allen Fällen wird die RPIC-Schaltung "Wartezustände" einführen, wenn bei Einleitung einer Übertragung keine Daten von der Quelle zur Verfügung stehen. Die konfigurierbaren Adreßmoduserweiterungen werden unter Verwendung der Steuersignale "Datenanforderung" an den Pipelinesteueranschlüssen implementiert.

Funktionsausführungsbefehle benutzen die Steuersignale "Funktionsanforderung" an den Pipelinesteueranschlüssen. Im allgemeinen gibt es für jeden PCP einen Funktionsanforderungsbefehl. Bei der RSP-Implementierung werden der konfigurierbaren Logik vier zusätzliche unbenutzte Befehlsoperationscodebits bereitgestellt, was die Implementierung von bis zu 128 verschiedenen von der konfigurierbaren Logik abhängigen Funktionen erlaubt.

Die PCP-Zustandsleitungen erlauben die Implementierung von "konfigurierbaren Sprung"-Befehlen. Damit können in Abhängigkeit von dem aktuellen Wert von Signalen in der konfigurierbaren Logik alternative Befehlsausführungsfolgen durchgeführt werden.

Konfigurierbare Unterbrechungen werden durch Erweiterung der typischen Unterbrechungsabbildungstabelle, so daß sie für jeden Pipeline-Steueranschluß eine unterbrechungsstelle enthalten, implementiert. Wenn die Unterbrechung von der konfigurierbaren Logik gesetzt ist, hält die RPIC die normale Programmausführung an und beginnt mit der Ausführung des Unterprogramms an der von einem Wert in der Unterbrechungstabelle definierten Adresse. Im normalen Gebrauch führt das Unterprogramm einen Funktionsanforderungsbefehl aus, wodurch die konfigurierbare Logikschaltung das Unterbrechungsanforderungssignal entfernt.

Eine Architektur für eine spezifische Implementierung kann direkte Ein-/Ausgabebefehle zur Implementierung von Datenübertragungen zwischen dem skalaren Prozessor, dem Schaltbus und den konfigurierbaren Logikmitteln benutzen. In einem typischen Fall werden den Pipelinebusanordnungsleitungssätzen Ein-/Ausgabe-"Anschluß"-Nummern zugewiesen. Diese Fähigkeit wird normalerweise beim Schaltbuskoppler benutzt, wenn dieses Mittel nicht als Parallelverarbeitungszusammenschaltung benutzt wird.

Beladen der mit den konfigurierbaren Logikkernzellen und der konfigurierbaren Verdrahtung verbundenen Konfigurationsregister wird durch Sonderbefehle durchgeführt. Jedes Register wird als eine Stelle im "Konfigurationsspeicher"-Adreßraum betrachtet. Ein Befehl des Typs "Blockverschiebung" wird zum Laden eines gesamten Pipelinesegments benutzt. Als Alternative kann jedes Konfigurationsregister durch eine Einzelbefehlsausführung geändert werden. Auch sind Befehle zum Lesen von einem oder mehreren Konfigurationsregistern vorgesehen. Diese Befehle werden zur Implementierung von adaptiven Algorithmen (beispielsweise adaptiven Filtern oder zur Änderung einer Pipelinestufe zur Durchführung einer "Addier"-Operation anstatt einer "Subtrahier"-Operation) benutzt. Wie bemerkt, kann die Quelle von Konfigurationsdaten ein chipinterner Speicher, chipexterner Speicher oder eine Schaltbusdateneingabe sein.

Die Fig. 16 ist ein Blockschaltbild mit dem detaillierten Datenfluß für die Hauptbuswege eines beispielhaften RSP-Bauelements. Der Lesedaten-(RD-read data)Bus hat zwei Quellen: (1) RDa von ALP 120 und (2) RDM vom Speicher mit Mehrfachzugriff 128. Die RPIC 122 steuert die Auswahl dieser alternativen Werte in Abhängigkeit von dem aktuellen Pipelinebetrieb. Gleichmaßen kann die Schreibdaten-(WD-write data)Quelle vom ALP oder PDP 126 kommen, die externen Schreib-(WX-write external)Daten können vom ALP oder dem PDP

kommen, die Leseadresse (RA-read address) kann vom ALP oder der RPIC kommen und die Schreibadresse (WA-write address) kann vom ALP oder der RPIC kommen. Zwischen dem Speicher und dem ALP ist ein besonders breiter Wortdatenweg für Konfigurationsdaten (CD) vorgesehen. Die Adreßerzeugungsschaltungen in der RPIC haben ihre eigene Verbindung zu den arithmetischen Pipelinedaten (F) im PDP. Gleichermäßen können die RPIC-Adreßregisterwerte unter Verwendung des Datenweges AR in den PDP-Berechnungen benutzt werden. Die RPIC liefert die Befehlsregisterinformationen zum ALP und auch die Pipelinesteueranschluß-(PCP-pipeline control port)Signale. Wenn als Hauptbussteuerung ein RSP-Bauelement benutzt wird, treibt die RPIC-Schaltung den C-Steuerbus, der einen Eingang für die RSP-Bauelemente darstellt, die als untergeordnete Datenverarbeitungselemente benutzt werden.

Schaltbus-Ein-/Ausgabeoperationen

Der Schaltbuskoppler (TBT-Toggle Bus Transceiver) wird als Hauptdateneingang und -ausgang für ein RSP-Bauelement benutzt. Wenn mehrere RSP-Bauelemente benutzt werden, um ein SIMD-Parallelverarbeitungssystem zu bilden, implementiert der TBT eine Vielzahl fortgeschrittener Datenflußmuster, die Signalfuß mit großer Bandbreite und parallele Verarbeitungsoperationen erlauben. Die vom TBT implementierten drei Arten von Datenfluß sind folgende:

Rundsenden — jeder Prozessor kann Daten an alle anderen Prozessoren rundsenden. Dies wird zur Befehlsverteilung vom Prozessor der Bushauptsteuerung an die untergeordneten Datenverarbeitungselemente und zur Datenrundsendung für die Schritte in einem Algorithmus, wenn jeder Prozessor dieselben Daten benötigt, benutzt.

Reflexionen — die Daten von Prozessor J werden zum Prozessor mit der Nummer $J \text{ XOR } R$ übertragen, wobei R der Hammingabstand ist. Reflexionen können auf Bitebene durchgeführt werden, um mehrdimensionalen Speicherzugriff zu unterstützen, der für Bildverarbeitungs- und Bilderzeugungsanwendungen benötigt wird.

Rotierungen — die Daten vom Prozessor J werden zum Prozessor mit der Nummer $(J + R) \text{ mod } N$ übertragen. Damit können ringbasierende Algorithmen veränderlicher Rotationsgröße implementiert werden. Rotationen können auch auf Bitebene durchgeführt werden, um Arithmetik- und Bilddatenhandhabung mit langen Worten zu unterstützen.

Jede dieser Operationen kann in einem einzigen Taktzyklus durchgeführt werden. Die Rundsendeoperationen benutzen einen Kombinationsdatenfluß wie bei einem herkömmlichen Tristate-Bus. Die Reflexions- und Rotationsoperationen benutzen ein zweiphasiges Schema, bei dem die Daten zuerst auf den P-Busleitungen übertragen, auf den Q-Leitungen empfangen und in einem Speicherregister gespeichert und danach auf den P- oder Q-Leitungen weiter übertragen werden. Zusätzlich zu den grundlegenden Datenflußoperationen führt der Schaltbus eine Anzahl von Overheadübertragungen durch, die Assoziativanordnungsverarbeitung erlauben. In der Tabelle 8 sind die Grundarten der Schaltbus-Datenübertragung zusammengefaßt.

Tabelle 8

Schaltbus-Datenflußoperationen

Name	Bus-Code	Abstand	Datenflußfunktion
NOP	0	-	Keine Operation; zur Prozessorsynchronisierung benutzt
CFG	1	0	Konfigurationsdatenrundsending von Bushauptsteuerung
DAT	1	1	Datenrundsending von Bushauptsteuerung für Datenverarbeitung
FCT	1	2	Befehlsrundsending von Bushauptsteuerung
GSEL	1	3	Gruppenauswahl
GSTAT	1	4	Gruppenstatuszusammenfassung
-	1	> 4	Reserviert
SEL	2	G	Bushauptsteuerung wählt Prozessoren der Gruppe G zur Aktivierung aus
STAT	3	G	Bus liest Prozessor-Statuszusammenfassung von Gruppe G
REF	4	D	Reflexion von Datenfluß mit Abstand D
BC	5	D	Rundsending von Daten vom Prozessor D
ROT	6	D	Rotieren von Datenfluß mit Abstand D
-	7	-	Reserviert

Der Leerlauf-Buszyklus "NOP" (no operation) erlaubt der Bushauptsteuerung die Synchronisierung einer Datenverarbeitungsanordnung. Eine mit "besetzt" bezeichnete Busleitung mit offenem Kollektor ist als Teil des Schaltbussteuervektors mit allen Prozessoren verbunden. Dieses Signal weist einen Wert "0" zur Bushauptsteuerung auf, wenn einer der Prozessoren gegenwärtig nicht zur Teilnahme an einem Schaltbuszyklus bereit ist. Ausführung des NOP-Buszyklus hält die Bushauptsteuerungsverarbeitung solange auf, bis alle Prozessoren bereit sind, weiterzuarbeiten. Der NOP-Zyklus kann von den Datenprozessoren zur Synchronisierung von internen Operationen benutzt werden.

Mit der Konfigurationsdatenrundsending "CFG" kann eine Bushauptsteuerung die ALP-Schaltungen eines oder mehrerer Datenprozessoren konfigurieren. Auf die CFG-Konfigurationsbuszyklen reagieren nur gegenwärtig aktive Prozessoren nach untenstehender Definition.

Der Datenrundsendedatenfluß "DAT" erlaubt der Bushauptsteuerung das Rundsending eines Datenwortes an alle Datenprozessoren in einer Anordnung. Die Datenprozessoren lesen diese Daten unter Verwendung von Befehlen "IN" oder durch selbständigen Pipelinezugriff. Der Funktionsrundsendingbuszyklus "FCT" erlaubt der Bushauptsteuerung die Übertragung eines Befehls an alle Datenprozessoren. Der Befehl kann eine Datenhandhabungsfunktion (wie beispielsweise ADD zum Akkumulator) oder eine Flußsteuerungsoperation (wie beispielsweise Springen zu einer unterprogrammstelle) sein. Wenn mehr als ein Wort erforderlich ist, können mehrere FCT-Buszyklen benutzt werden.

Der Auswahldatenfluß "SEL" erlaubt dem Prozessor der Bussteuerung die Auswahl einer Teilmenge der Datenprozessoren für nachfolgende Operationen. Für 1 bis 16 Prozessoren entspricht jedes Datenbit in der Rundsending dem "Aktivitäts"-Zustand des entsprechenden Prozessors. Beispielsweise setzt das Bit 0 der Daten den aktiven Zustand für den Prozessor 0 bzw. setzt ihn zurück. Wenn es mehr als 16 Prozessoren gibt, erlaubt der Abstandswert "G" die Auswahl einer Teilmenge von 16 Prozessoren, die eine Gruppennummer aufweisen, die dem Wert von "G" entspricht. Wenn der Aktivitätszustand gesetzt ist, reagiert ein Prozessor auf die Funktionsrundsendingbuszyklen "FCT". Wenn der Aktivitätszustand "0" ist, führt ein Prozessor bei Ausführung eines Funktionsbuszyklus keine Operation durch. Der Gruppenzustands (GSTAT-Group Status) Zyklus sendet der Bushauptsteuerung ein 16-Bit-Wort zurück, wobei jedes Bit eine Zusammenfassung einer Teilmenge von 16 Prozessoren ist. Ein Bit ist eine "1" wenn einer der Prozessoren in der Gruppe ein Zustandsbit von "1" aufweist.

Der Lesezustandsbuszyklus "STAT" wird zum Lesen eines Zusammenfassungszustandswortes aus einer Gruppe von Datenprozessoren benutzt. Jedes Bit in dem von der Bushauptsteuerung empfangenen Wort entspricht einem zusammenfassenden Zustandsbit in jedem Prozessor. Die Bedeutung des Zustandsbits ist vom Algorithmus abhängig. Wenn es mehr als 16 Prozessoren gibt, wählt der Parameter "G" die Gruppe aus, die auf die Statusanforderung anspricht. Die Statusantwort wird typischerweise in Verbindung mit dem Auswahlbuszyklus für Operationen des Algorithmus der Assoziativanordnung benutzt.

Fig. 17 ist ein Blockschaltbild einer beispielhaften Schaltung des Schaltbuskopplers 124 für ein RSP-Bauelement. Die interne Datenquelle des RSP für die Datenausgabe sind entweder der Bus WXa von der ALP-Einheit oder die wxpDaten von der PDP-Einheit. Vom Schaltbus empfangene Daten werden auf dem Bus RX verteilt.

Bei einer Rundsendeoperation werden die WX-Daten nach Umwandlung in der Schaltbusaustauschschaltung 260 über die Ansteuerschaltung 270 übertragen. Die Daten werden dann von den P-Leitungen über P-Speicher 262 und RX-Multiplexer 264 zur Verteilung auf dem RX-Bus empfangen. Für Reflexions- und Rotationsoperationen werden die Daten nach Umwandlung zuerst über den P-Treiber 266 angesteuert. Die Daten werden dann im Q-Speicher 268 empfangen und gespeichert. Bei der zweiten Taktphase werden die Q-Speicherdaten dann über die Umwandlung zur Übertragung zurück durch P-Treiber 266 oder Q-Treiber 270 geführt und dann abschließend vom Q-Speicher 268 oder P-Speicher 262 aus auf den RX-Multiplexer 264 aufgeschaltet. Die P- und Q-Speicher fungieren als Master-Slave-Flipflop zum Speichern der Empfangsdaten für Reflexionen und Rotationen.

Der Schaltbuskoppler 124 wird durch den C-Vektor gesteuert, der folgende Informationen liefert:

- Schaltbusphase;
- P- und Q-Treiberfreigaben;
- Aktueller Datenflußtyp; Rundsenden, Reflexion, Rotation usw.; und
- Abstand: Rundsendequelle, Reflexionsabstand oder Rotationsabstand.

Der Schaltbuskoppler 124 erlaubt einen unterschiedlichen Datenfluß mit jeder Phase eines Buszyklus. Beispielsweise können Daten auf Bitebene mit der ersten Phase rotiert und dann mit der zweiten Phase reflektiert werden.

Bei SIND-Systemen erzeugt der Prozessor der Bushauptsteuerung die Komponenten des C-Vektors für alle Datenprozessoren. Der C-Vektor wird dann in die untergeordneten Prozessoren eingegeben. Beim Einschalten fungiert der Schaltbus nur als Rundsendebus, bis alle RSP-Bauelemente erstkonfiguriert sind.

Detaillierte Beschreibung der RPIC

Die vollständige Schaltung für die rekonfigurierbare Pipeline-Befehlssteuerung (RPIC-Reconfigurable Pipeline Instruction controller) ist von der Implementierung des Pipeline-Datenprozessors (PDP) abhängig. Die Schnittstelle zwischen dem RPIC-Block und dem ALP ändert sich nur mit der in der Pipeline-Busanordnung enthaltenen Anzahl von Datenwegen und der Anzahl von in den Pipeline-Steueranschlüssen implementierten Signalen. Die Schnittstelle zwischen dem RPIC-Block und dem Kernprozessor, dem Speicher und den Parallelverarbeitungs-Zusammenschaltfestfunktionen ändert sich nach den genauen Erfordernissen der zur Implementierung dieser Festfunktionsblöcke gewählten Schaltungen.

Wenn der ALP mit einem Kernprozessor verbunden ist, dann benötigt der RPIC-Block Steuersignale, die folgendes anzeigen: (1) wann ein neuer Befehl von einem Programmspeicher geholt wird, (2) wann vom Prozessor eingeleitete Ein- und Ausgabezyklen auszuführen sind und (3) wann Daten zu und von einem Festfunktionsdatenspeicherblock zu lesen bzw. schreiben sind. Auch erwartet die RPIC-Schaltung, daß sie Softwareunterbrechungsfunktionen im Kernprozessor einleiten kann. Diese generischen Fähigkeiten stehen bei allen Prozessoren zur Verfügung, ändern sich aber in den spezifischen Einzelheiten der Signale und in der Zeit der Signale.

Wenn der ALP mit einem Festfunktions-Parallelverarbeitungszusammenschaltblock verbunden ist, dann benötigt der RPIC-Block Schnittstellensteuersignale für folgendes: (1) zum Bestimmen, wann eine Rekonfiguration von einer externen Vorrichtung angefordert wird, (2) Bestimmen, wann extern eingeleitete Befehle von der RPIC auszuführen sind, (3) Bestimmen, wann externe Daten dem ALP verfügbar gemacht worden sind und (4) Bestimmen, wann die externen Vorrichtungen zur Aufnahme von Daten vom ALP bereit sind.

Wenn der ALP mit einem Festfunktionsspeichermittel verbunden ist, dann benötigt die RPIC Schnittstellensteuersignale zum (1) Bestimmen, wann der externe Speicher zum Schreiben zur Verfügung steht und (2) Bestimmen, wann vom ALP angeforderte Daten zur Verwendung bereitstehen.

Die folgende Beschreibung beschreibt ein minimales RPIC-Teilsystem, wobei ein generischer PDP mit Betonung auf der Schnittstelle zu den Pipeline-Steueranschlüssen (PCP-pipeline control ports) angenommen wird.

Fig. 18 ist ein Blockschaltbild mit den Grundbauelementen einer RPIC-Schaltung 122. Die RPIC enthält zwei Grundregister für den Programmsteuerfluß. Befehlsregister (IR-instruction register) 280 hält den aktuellen auszuführenden Befehl und Programmzähler (PC-program counter) 282 hält die Adresse des nächsten auszuführenden Befehls. IR 280 kann entweder vom lokalen Speicher unter Verwendung der Adresse im PC 282 oder vom Empfangsanschluß des Schaltbusses aus beladen werden. Der Schaltbuseingang wird zum Empfang der Befehle in einem Datenprozessor von einem Bushauptsteuerungsprozessor unter Verwendung des Funktions-Schaltbuszyklus "FCT" benutzt. Die externe Befehlseingabe fungiert als "weiche Unterbrechung". Das heißt, es wird jede laufende Programmausführung während der Ausführung des externen Befehls (der normalerweise ein Abrufbefehl für ein Unterbrechungsdienstprogramm ist) angehalten. Die Ausführung eines Unterbrechungsrücksprungbefehls bewirkt die Wiederaufnahme der Ausführung durch die RPIC an der Stelle im Programm, an

der die Funktionseingabe stattfand.

Der Programmzähler 282 steuert den Fluß des RSP-Programms durch die Abfolge von Adressen auf dem Programmadreßbus (PA). Für die meisten Befehle erhöht sich der PC-Wert mit demselben Taktzyklus, mit dem das IR beladen wird. Zur direkten Veränderung des aktuellen PC-Wertes werden Sprung- und Abrufbefehle benutzt.

Adressen zum Speicherlesen (auf dem Leseadreßbus RA) kommen von vier möglichen Quellen: (1) dem Speicherlesebus (RD), der als das zweite Wort eines Befehls eine direkte Adresse im Programmspeicher ist, (2) dem Schaltbus als zweites Wort einer Schaltbusfunktionseingabe, (3) dem ALP-Pipelinebus RAa als Reaktion auf eine selbständige Pipelineanforderung (ARQ) oder von einer durch Befehl eingeleiteten konfigurierbaren Adreßmodusoperation und (4) dem Prozessor indirekter Adressen (IAP-indirect address processor) 284. Schreibadressen für den Speicher auf dem WA haben gleichartige Quellwerte.

Der Prozessor indirekter Adressen 284 enthält PDP-abhängige Adreßregister- und Adreßberechnungsmittel zur Erstellung von Adressen auf Basis von Auswahlbit für indirekten Adreßmodus im Befehl.

Befehlsdecodierschaltung (IDC-Instruction Decode Circuit) 286 bestimmt die Art der Adressierung für jeden Befehl und wählt die Adreßquelle für RA und WA aus. Die Konfigurationsregister (CR) 288 in der RPIC enthalten Daten, die den Kontext der RSP-Operationen definieren. Beispielsweise werden CR-Bits zur Kennzeichnung der Prozessorgebrauchsart (Bushauptsteuerungsprozessor oder Datenprozessor) benutzt. Das CR enthält auch die Pipeline-Busanordnungskonfigurationssteuersignale wie schon beschrieben.

Die Pipeline-Vergabeschaltung (PARB) 290 koordiniert den Betrieb zwischen den ALP-Pipelinebussen, der Befehlsabrufpipeline und den PDP-Datenverarbeitungspipelines. Das benutzte Schema basiert auf der herkömmlichen Philosophie verriegelter Pipelineverzögerung: wenn eine Pipeline Daten von einer zweiten Pipeline benötigt, wird die erste Pipeline solange aufgehalten (ihr Taktsignal solange unterdrückt), bis die Daten von der angeforderten Pipeline zur Verfügung stehen. Die Software und ALP-Schaltungen sind dafür verantwortlich, ordnungsgemäßen Pipelinefluß sicherzustellen, indem sie nach Bedarf NOP-Befehle bzw. Taktwartezyklen einfügen.

Im einzelnen ist die Implementierung der PARB von der PDP-Struktur und von der Art der benutzten Technik abhängig. Die folgenden Absätze beschreiben die Funktionsweise der PARB-Schaltung im Zusammenhang mit den Grundoperationen, an denen die konfigurierbaren Pipelineschaltungen des ALP beteiligt sind.

Pipeline-Freigabeoperation — vom Programm eingeleitete und selbständige Datenübertragungen zwischen dem ALP und den anderen RSP-Prozessoren unter Verwendung der Pipeline-Busanordnung erfordern, daß die Pipeline durch folgendes freigegeben wird: (1) das Konfigurationsregister aktiv (CRAj) und (2) das Aktivsignal (ACTj) vom ALP.

Fig. 19 ist ein Zustandsdiagramm, das den Funktionsfluß zur Erzeugung eines Pipelinefreigabesignals für das eingeleitete Programm und selbständige Datenübertragungsoperationen zeigt. Die zur Implementierung des gezeigten Zustandsdiagramms benötigten Schaltungen können leicht von einem Fachmann hergestellt werden. Beim Einschalten und nach einem harten Rücksetzen werden alle Pipelines in den Zustand "gesperrt" getrieben. Damit werden die Pipelinefreigabesignale (PENj) auf LOW gezwungen, wodurch die anderen Operationen gesperrt werden. Die Pipeline bleibt solange gesperrt, wie eine Konfigurationsfolge ausgeführt wird (Signal CFG ist HI) und das Steuerbit Konfigurationsregisteraktiv LOW ist. Wenn CRAj HI ist und keine Konfigurationsfolge ausgeführt wird, überprüft die Schaltung das Signal ACTj vom ALP im Zustand "Überprüfen". Wenn ACTj auf LOW gesetzt ist, tritt die Schaltung in den Zustand "Freigegeben" ein und erzeugt einen Wert HI beim Signal PENj für die Pipeline. Eine Pipeline wird dann gesperrt sein, wenn eine andere Konfigurationsfolge ausgeführt wird oder eine Operation durchgeführt wird, die das Signal ACTj auslöst.

Fluß der Funktionsanforderungsoperation — Fig. 20 ist ein Zustandsdiagramm, das den Betriebsfluß zeigt, wenn der Befehlsdecoder bestimmt, daß der aktuelle Befehl Ausführung einer Pipelinefunktion im ALP erfordert. Wenn die angeforderte Pipeline nicht im Konfigurationsregister freigegeben ist oder das Aktivsignal (ACTj) für die Pipeline nicht gesetzt ist, wird eine Fehlerunterbrechung eingeleitet. Damit wird die Fehlersuche in Algorithmen vereinfacht, die ALP-Pipelineschaltungen benutzen. Andernfalls gibt der RPIC-Befehlsdecoder das Signal "Programmfunktionsanforderung" (PFRQj) an den Programmfunktionszustandsautomaten für die Pipeline aus. Wenn das Signal "Funktion bereit" nicht gesetzt ist (das Signal FRYj ist HI), wird in einen Zustand beschäftigt eingegangen, wobei das Signal "Funktion anfordern" (FRQj) auf LOW gesetzt ist. Damit wird allen nachfolgenden Funktionsanforderungen an diese Pipeline ein Aufhaltezustand aufgezwungen. Die Schaltung verbleibt im Zustand "beschäftigt", bis das Signal FRYj auf LOW gesetzt wird oder die Pipeline gesperrt wird.

Solange das Signal FRYj gesetzt bleibt, kann mit jedem Taktzyklus ein neuer Pipeline-Funktionsbefehl ausgeführt werden. Dies ermöglicht die Implementierung hochratiger Pipelines, die in der Lage sind, mit jedem Taktzyklus eine neue Operation zu beginnen. Es erlaubt auch die Implementierung von Pipelines, die mehr als einen Taktzyklus für jede Funktionsanforderung erfordern. Beispielsweise wird eine Pipeline, die für jede Funktionseinleitung zwei Taktzyklen erfordert, das Signal FRYj einen Taktzyklus lang nach Erkennung des Signals FRQj auf HI anheben. Die PARB setzt dann die Funktionsanforderung (FRQj) für die ausgewählte Pipeline und wartet auf das Setzen des Zustands Funktion bereit (FRYj) von der ALP-Schaltung. Wenn im Befehlsstrom kein Pipeline-Funktionsbefehl anhängig ist, löst die PARB die Funktionsanforderung aus.

Fluß der Datenanforderungsoperation — Das RSP-Programm kann Befehle ausführen, die explizit die Datenübertragung von einer Verarbeitungseinheit (Speicher, PDP oder Schaltbus) oder von einer ALP-Pipeline zu einer anderen Verarbeitungseinheit vereinlassen. Diese Befehle bedienen sich der "Datenanforderungs"-Synchronisationschaltung. Auch kann der Befehlsdecoder in der RPIC einen Datenanforderungszyklus für die konfigurierbaren Adreßmodi einleiten, die die Verwendung einer ALP-Pipelineausgabe für die Lese- oder Schreibadresse erlauben. Konfigurierbare Adreßmodi leiten auch eine Datenanforderung zur Übertragung von Daten vom PDP zu einer ALP-Pipeline oder vom ALP zum PDP ein. Alle diese Datenflußoperationen bedienen

sich der Steuersignale-datenanforderung (DRQj) und Daten bereit (DRYj) am Pipelinesteueranschluß.

Die Fig. 21 ist ein Zustandsdiagramm, das den Funktionsfluß der Programmdatenanforderungsoperation zeigt. Wie bei allen vom Programm eingeleiteten Operationen muß die angeforderte Pipeline freigegeben werden, sonst wird eine Fehlerunterbrechung ausgeführt. Die Datenanforderung (DRQj) wird von der PARB-Schaltung für den ALP gesetzt, wenn der Befehlsdecoder das Signal Programmdatenanforderung" (PDRQj) ausgibt. Wenn die Pipeline nicht für die Datenübertragung bereit ist, tritt der Zustandsautomat in den Zustand "beschäftigt" ein und wartet darauf, daß das Signal "Daten bereit" (DRYj) auf LOW gesetzt wird. Im Zustand beschäftigt wird eine nachfolgende Programmdatenanforderung die Befehlsausführungspipeline so lange aufhalten, bis das Signal DRYj auf LOW gesetzt wird. Mit jedem Taktzyklus kann so lange eine neue Datenanforderung gesetzt werden, wie die ALP-Pipelineschaltung in der Lage ist, Datenübertragung mit der höchsten Geschwindigkeit aufrechtzuerhalten. Ansonsten kann die ALP-Steuerschaltung "Wartezustände" einfügen, indem sie DRYj für die Dauer dieser notwendigen Anzahl von Taktzyklen auf HI setzt. Die RPIC fügt Wartezustände in den Vorgang ein, wenn die Quellen- bzw. Zielverarbeitungseinheit zusätzliche Zeit für die Datenübertragung erfordert.

Fluß der Statuszustandsoperation — Die ALP-Pipeline-Zustandssignale (STATj) werden von den bedingten Sprungbefehlen des RSP gelesen, die die ALP-Pipeline als eine Sprungbedingung ansprechen. Im Bezug auf die anderen vom Programm eingeleiteten Operationen wird eine Fehlerunterbrechung erzeugt, wenn die Pipeline nicht freigegeben ist. Die RPIC liest einfach die Statusleitung ab und führt den Sprung durch, wenn der Zustand erfüllt ist (Sprung mit Status "0" oder Sprung mit Status "1"). Man beachte, daß es für diese Funktion keinen Anforderungs/Bestätigungs-Austausch gibt. Der Algorithmus muß Aufhaltezyklen einfügen oder einen ALP-Funktionsbefehl ausführen, der darauf wartet, daß der Zustand vor Ausführung des bedingten Sprungs gültig ist, um sicherzustellen, daß der Zustand bei Ausführung des bedingten Sprunges gültig ist. Die meisten ALP-Pipelines halten das Signal STATj zu allen Zeiten auf einem gültigen Wert.

Selbständige Pipelinefunktion — Wenn sie in den Konfigurationsregistern freigegeben ist, kann eine ALP-Pipeline eine selbständige Datenübertragung zwischen der ALP-Pipeline-Busanordnung und dem Schaltbus oder den RSP-Speichermitteln anfordern. Diese Übertragungen erfordern keine Ausführung eines RSP-Befehls, sind aber normalerweise durch Funktionsanforderung und Datenanforderungsbefehlsausführung mit der Programmausführung synchronisiert. Eine selbständige Operation bedient sich der Signale selbständige Anforderung (ARQj) und selbständiges Bereit (ARYj) an einem Pipeline-Steueranschluß.

Die Fig. 22 ist ein Zustandsdiagramm, das den Funktionsfluß der selbständigen Anforderungsfolgeoperation zeigt. Die Pipeline muß im Konfigurationsregister für selbständige Handlung freigegeben sein, wenn eine selbständige Anforderung (ARQj) erkannt werden soll. Wenn ein Anforderungssignal gesetzt ist und die Pipeline nicht freigegeben ist, wird kein Fehlerunterbrechungssignal erzeugt.

Zur eingehenderen Erläuterung der selbständigen Pipelineoperation wird in der folgenden Besprechung eine selbständige Anforderung an das RSP-Speichermittel angenommen. Die Adresse für diese Anforderung wird von der RA-Pipeline im ALP geliefert. Es wird angenommen, daß die Adresse bei Setzen der Anforderung gültig ist. Die PRB-Schaltung setzt eine Leseanforderung an den Speicher und setzt das Signal bereit (ARYj) auf LOW, wenn die Daten verfügbar sind (Signal PRYk wird von der Speichereinheit auf LOW gesetzt). Wenn die Daten von der DRm-Pipeline angenommen worden sind, entfernt die ALP-Schaltung die Anforderung (ARQj) und die PARB reagiert mit der Entfernung des Signals bereit (ARYj).

Für jeden Zyklus, in dem das Anforderungssignal gesetzt ist und das Signal bereit gesetzt ist, wird eine Datenübertragung durchgeführt. Damit können schnelle Pipelines mit jedem Taktzyklus eine Speicheransteuerung unternehmen. Wenn eine Folge asynchroner Anforderungen die Speicheranforderungen für den PDP oder Schaltbus stört, wird der ALP-Pipeline nur die Hälfte der verfügbaren Speicherzyklen erteilt. Dies tritt nur bei gleichzeitigen Leseoperationen und gleichzeitigen Schreiboperationen ein.

Eine selbständige Schreiboperation zum RSP-Speicher wird durch eine ähnliche Folge durchgeführt. In diesem Fall muß die Schreibadresse am Waa-Pipelineausgang gültig sein und die vom WDa-Bus zu schreibenden Daten müssen vor Stellen der Anforderung gültig sein. Die RPIC setzt ARYj, wenn die Daten für die Speicherschreiboperation angenommen worden sind.

Selbständige Übertragungen zur und vom Schaltbuskoppler werden auf ähnliche Weise durchgeführt. Eine Leseanforderung wird aufhalten, bis der Bushauptsteuerungsprozessor einen Taktbuszyklus vollendet hat. Eine Schreibenanforderung wird aufhalten, bis der Zyklus abgeschlossen ist, hat aber auch die Eigenschaft, daß zum anfordernden Prozessor übertragene Daten (beispielsweise bei Benutzung eines Rotierungszyklus) aus dem RX-Pipelineeingang ausgelesen werden können, wenn das Signal ARYj gesetzt ist. So kann ein normaler Schreibbus-Schreib-/Lesezyklus mit einer einzigen selbständigen Anforderung durchgeführt werden.

Konzepte der Konfigurationsfolge

Der ALP in einem RSP-Bauelement kann Konfigurationsdaten von einer von drei Quellen empfangen: (1) externe Daten (auf dem RX-Bus vom Schaltbuskoppler), die ihren Ursprung in einem Bushauptsteuerungs-RSP-Bauelement haben, (2) externe Daten von einem Speicher oder Systembus oder (3) von dem Speicher mit Mehrfachzugriff. Beim Einschalten zeigt ein Eingabeanschluß an, daß ein RSP unter Verwendung von externen Daten zu konfigurieren ist oder daß der RSP einen externen Speicher adressieren soll, um die Konfigurationsdaten zu erhalten. Konfiguration des ALPs aus dem chipinternen Speicher bietet einen schnellen Mechanismus für rekonfigurierbare Pipelineänderungen, erfordert aber, daß der Speicher zuerst von einer externen Quelle beladen wird.

Die allgemeine Konfigurationsdatei für die RSP-Bauelemente enthält größenveränderliche Datensätze für sechs verschiedene Mengen rekonfigurierbarer Speicher und Register:

Kernzellenanordnung (CCA-core cell array) — Erlaubt Rekonfiguration eines beliebigen rechteckigen Nachbarblocks der ALP-Anordnung. Jede Zelle erfordert ein Wort (zwei Byte) Konfigurationsdaten. Für die RSP-Familie wird die ALP-Anordnung als lineares Abbild ähnlich der für Bitmap-Graphiken benutzten Art adressiert. Das heißt, es wird angenommen, daß die Anordnung eine Größe von N_c Spalten (Taktleitungsrichtung) und N_r Zeilen (Datenwegrichtung) besitzt. Die Adresse für die Zelle an Stelle x (Spalte) und y (Zeile) ist dann gegeben durch: $A = x + y \cdot N_r$. Eine gegebene Adresse löst sich durch die folgenden Formeln in x - und y -Koordinaten auf: $x = A \bmod N_r$ und $y = A \div N_r$. Zwecks Verträglichkeit mit der CLAY-Architektur steigt die x -Koordinate von links nach rechts an und die y -Koordinate steigt von unten nach oben an, so daß Adresse 0 die untere linke Ecke des Blocks darstellt. Mit diesem Schema können Konfigurationsblöcke unabhängig von den Anordnungsabmessungen sein. Ein Kernzellenanordnungskonfigurationsblock wird dann durch die Zieladresse (unter linke Ecke des Blocks in der Zielanordnung), die ein Einzelwort ist, die Blockbreite (ein Byte) und die Blockhöhe (ein Byte) definiert.

Lokalbusanordnung (LBA-local bus array) — Der ALP behält die grobe Blockstruktur der CLAY-Bauelemente für lokale und Express-Buskonfiguration. Die Konfiguration für die horizontalen und vertikalen Verstärkerschaltungen (HR und VR — horizontal repeater, vertical repeater) für den Lokalbus wird wie bei den Kernzellen adressiert.

Rekonfigurierbare E/A-Anschlüsse (RIOP Reconfigurable I/O Pins) — die Peripherie der ALP-Anordnung enthält vom Familienmitglied abhängigen Zugang zu rekonfigurierbaren E/A-Anschlüssen sowie Zugang zu RSP-internen Steuersignalen. Damit kann die Peripherie der Anordnung unter Verwendung von "Vierer-Vierer"-Signalen auf Nachbaranordnungen erweitert werden. In dem RSP-Fall hat jeder E/A-Anschluß Zugang zu RSP-internen Signalen wie auch die gewöhnliche Bedeutung rekonfigurierbarer Anschlußstellen. Die Adressierung von Peripheriezellen ist notwendigerweise von der Vorrichtung abhängig. Es gibt mehrere Unterschiede im Verhalten der RIOP im ALP.

Pipeline-Busanordnung (PBA-Pipeline Bus Array) — Dies ist der bedeutendste Zusatz zur CLAY-Architektur und würde den meisten konfigurierbaren Logikauslegungen hinzugefügt werden müssen. Die RSP-Bauelemente enthalten eine Anzahl von Pipeline-Bussen zum Zugreifen auf den Daten-MPM, TBT und auf PDP-Daten und RPIC-Steuersignale. Die PBA ist eine Signalmenge mit doppelter Verwendung, die nur in der horizontalen Richtung fließt. Während einer Konfigurationsfolge wird die PBA zur Zuführung von Konfigurationsdaten zu den Kernzellen, Hr, VR und RIOP benutzt. Wenn die interne Anordnung nicht gerade konfiguriert wird, weist jede Anordnungszeile acht Leitungen auf, die direkt unter Verwendung der HR mit lokalen Bussegmenten verbindbar sind. Für jede Zeile weisen vier der PBA-Leitungen fest zugeordnete Verbindungen mit RSP-Pipelinebussen auf und vier der PBA-Leitungen werden als globale Verbindungsleitungen innerhalb des ALPs benutzt. Die PBA-Leitungen mit fest zugeordneter Bedeutung weisen Bitzuweisungen auf, die für Datenwegblöcke mit Raster 2 versetzt sind. Ein globales Konfigurationssignal CFG wird durchweg über ER verteilt, um lokale Bustreiber zu PBA-Leitungen während einer Konfigurationsfolge abzutrennen.

RSP-Konfigurationsregister (RSPCR-RSP Configuration Registers) — Der RSP enthält 32 Byte von SRAM, die der statischen Konfiguration der RSP-Umgebung zugeordnet sind. Diese Information enthält Bit für Prozessorgeschwindigkeit, Schaltbusgeschwindigkeit, Startprogrammadresse und so weiter.

Speicher mit Mehrfachzugriff (MPM-Multiple Port Memory) — Der Inhalt des internen Programmspeichers des RSPs ist als Teil einer Konfigurationsfolge konfigurierbar. Damit kann ein Erstprogramm ("Bootstrap-Lader") beim Einschalten oder nach einem Wieder-Urladen geladen werden.

Die verschiedenen Arten von Konfigurationsdatensätzen werden, wie in Tabelle 9 dargestellt, durch ein Huffman-Codierschema unterschieden.

Tabelle 9

Konfigurationsadreßliste

Adressbits	Konfigurationsmittel
00cc cccc cccc cccc	Kernzellenadresse
01bb bbbb bbbb bbbb	Lokalbusanordnung
1000 iiii iiii iiii	E/A-Anschlußstellenanordnung (Peripherieanordnung)
1001 gggg gggg gggg	Globalbusanordnung (sofern benötigt)
1010 ---- -r rrrr	Steuerregisterkonfiguration
1011	(reserviert)
11aa aaaa aaaa aaaa	Programmspeicher

Mit diesem Schema wird die direkte Abbildung für eine CCA von bis zu 16.384 Zellen (128mal 128) unterstützt. Größere RSP-Teile würden 32- oder 64-Bitworte aufweisen, womit größere Anordnungen unterstützt werden.

Jeder RSP-Teil unterstützt die in Tabelle 10 gezeigten Befehle zur Rekonfiguration.

Der Befehl BCON muß vor Ausführung der Befehle DYCON, LDCON und STCON ausgeführt werden.

- 5 BCON setzt den ALP-Taktmodus auf die durch die letzte Konfiguration gesetzten Werte und legt das Signal CFG an die ER-Verbindungen zur PBA an. DYCON führt eine Einzelblockkonfiguration durch. Die Zieladresse, Breiten- und Höheninformationen werden als die ersten zwei Worte des Konfigurationsblocks im Speicher definiert. Die Befehle LDCON und STCON erlauben das Lesen und Schreiben von Einzelzellenkonfigurations-
- 10 STCON befinden sich im Konfigurationsadreßraum, während die "Adresse" für REBOOT und DYCON sich im Programmspeicheradreßraum befindet.

Tabelle 10

RSP-Konfigurationsbefehle

Befehl	Format	Funktion
REBOOT	op I Adresse	Ausführen Wieder- Urladefolge unde Beginnen Konfigurations-datei- Stromfolge mit Anfang bei Speicheradresse
BCON	op	Beginnen Konfigurierung: PBA abrennen Konfigu- rations-taktmode
DYCON	op I Adresse	Teil-Rekonfigurieren Block aus Speicheradresse
LDCON	op I Adresse	Laden Konfigurationswort aus Konfigurations- adreßraum in Akkumulator
STCON	op I Adresse	Setzen Konfigurationswort aus Akkumulatorregister
ECON	op	Enden Konfigurierung: Freigeben ALP PBA, Takte und E/A

Konfigurationsdateistruktur

- 55 Eine Konfigurationsdatei besteht aus einem Kopf, einer Menge von Konfigurationsblöcken und einem Nachspann. Die allgemeine Struktur gleicht den CLAY-Bitströmen mit folgenden Unterschieden:

- RSP-Vorrichtungen lassen sich nur im Parallelmodus konfigurieren und es besteht daher keine Notwendigkeit für Bitsynchronisierung. Eine RSP-Vorrichtung kann anfangs aus externen Daten (externer Taktmodus) oder durch interne Adressenerzeugung (interner Taktmodus) konfiguriert werden;
- 60 — Das erste Byte einer Konfigurationsdatei besteht aus einem Vier-Bit-"Vorspann"-Muster (nominell 1011) und einem Vier-Bit-Konfigurationsdateisteuerfeld. Das LSB des Steuerfeldes definiert die Konfigurationsquellspeicherbreite (0 → acht Bit breiter Speicher und 1 → 16-Bit-breiter Speicher). Die übrigen drei Bit definieren die Speichergeschwindigkeit (Anzahl von internen Taktzyklen pro externer Speicheransteuerung). Damit kann ein RSP sich selbst von einer Vielzahl von externen ROM- oder FLASE-Speichervorrichtungen aus konfigurieren. Im externen Konfigurationsmodus definieren die Steuerbit die Quelldatenbreite;
- 65 — Das zweite Byte einer Konfigurationsdatei (das hochwertige Byte des ersten Wortes bei einer 16-Bit-Breite) erweitert die Konfigurationsdateisteuersignale. Dieses Byte enthält Taktsteuerinformationen;
- Das zweite Wort einer Konfigurationsdatei zeigt die Anzahl von Konfigurationsblockdatensätzen an, die

folgen. Es besteht keine Notwendigkeit für eine "Konfigurationssprungadresse", da diese Funktion leicht durch das aktuelle aktive Programm durchgeführt wird;

— Jeder Konfigurationsblock besteht aus einem Zwei-Wort-Kopf, gefolgt von den Konfigurationsdaten (stets ein Mehrfaches von zwei Byte). Das erste Kopfwort ist die Zielkonfigurationsadresse. Das niederwertige Byte des zweiten Wortes ist die Blockbreite und das hochwertige Byte des zweiten Wortes ist die Blockhöhe;

— Eine Konfigurationsdatei endet mit einem Nachspannbyte, das dasselbe wie das bei den CLAY-Vorrichtungen benutzte ist. Es gibt kein Konzept für "Kaskade". Wenn in einem System mehrere RSP-Teile (oder ein RSP mit mehreren CLAY-Vorrichtungen) benutzt werden, dann kann ein Haupt-RSP andere Vorrichtungen unter RSP-Programmsteuerung konfigurieren.

In der Tabelle 11 ist die RSP-Konfigurationsblockstruktur zusammengefaßt, während in der Tabelle 12 die RSP-Konfigurationsdateistruktur zusammengefaßt ist.

Tabelle 11

Konfigurationsblockdatensatzstruktur

Wort	Format	Bedeutung
0	W	Zielkonfigurationsadresse
1	H,L	Höhenbyte, Breitenbyte
2	W	H * L Datenworte

Tabelle 12

Konfigurationsdateidatensatzstruktur

Wort	Format	Bedeutung
0	H, [r,c]	Vorspannwort: Niederwertiges Byte enthält Vorspannmuster und Wortgrößenbit. Höherwertiges Byte enthält Konfigurationssteuersignale
1	W	Anzahl von Konfigurationsblöcken
---	Blöcke	Konfigurationsblockdatensätze
letztes	W	NACHSPANNwort

Externe Konfiguration bedient sich stets der Konfigurationsdateidatensatzstruktur. Jeder DYCON-Befehl zur internen Konfiguration steuert nur einen einzigen Block an. Konfigurationssteuersignale für einen DYCON-Befehl werden durch die RSPCR-Registerbit definiert, die durch Ausführung von STCON-Befehlen geändert werden können.

Mehrfache RSP-Konfigurationen

Wie erwähnt, können mehrere RSP-Vorrichtungen miteinander unter Verwendung der Schaltbusverbindungsarchitektur verbunden werden, um die Implementierung einer Anzahl von Arten eines Parallelverarbeitungssystems zu erlauben. Die allgemeinste Struktur ist die in Fig. 1B dargestellte. Wie in der vorherigen Besprechung dieser Figur bemerkt, wird ein als Prozessor "P_N" bezeichneter Prozessor als Bushauptsteuerungs-

prozessor für eine Gruppe von N Verarbeitungselementen, die als Prozessoren " P_0 " bis " P_{N-1} " bezeichnet werden, benutzt. Bei der gegenwärtigen RSP-Implementierung ist der Schaltbus auf $N \leq 256$ begrenzt. Jeder Schaltbuskoppler besitzt zwei, "P"-Bus und "Q"-Bus genannte, zweiseitig gerichtete M-Bit-Koppler. Jeder Schaltbuszyklus besteht aus der Übertragung der Datenausgabe über die Leitungen "P" oder "Q" und dem Empfang der Daten auf den alternativen Leitungen "Q" bzw. "P". Der gemeinsame Schaltbus-Steuerbus "C" wird zur Definition der Art der Bustransaktion für alle Prozessoren benutzt.

Der Schaltbus bietet eine "Breitwort"-Fähigkeit, die bei vorherigen Parallelverarbeitungssystemen nicht verfügbar ist. Für M-Bit-Worte pro RSP-Knoten und N Knoten können die MAN Bit auf dem Schaltbus durch den Compiler als einziges Breitwort angesehen werden. Zu beispielhaften Operationen, die an diesem Breitwort in einem Buszyklus durchgeführt werden können, gehört "Rotieren um k ", wobei $k \leq M \cdot N$, und die Bitumkehr des vollständigen Breitwortes. Diese Fähigkeit hat zahlreiche Anwendungen, einschließlich einer Arithmetik mit sehr hoher Präzision, die Implementierung von Fähigkeiten einer Arithmetik des finiten Feldes und die effektive Lösung einer Anzahl von Problemen, die bei Logikoptimierung und -analyse auftreten.

Die in der Figur aufgeführte "Schaltbusverdrahtung" basiert auf einem Muster der "idealen Mischung". Für den Fall von M-Bit-Worten und $N = M$ Prozessoren wird die Verdrahtungsregel wie folgt definiert. P_{kj} stelle die j -te Bitleitung des P-Busses am Prozessor k dar und Q_i stelle gleichermaßen die i -te Bitleitung des Q-Busses an dem mit i numerierten Prozessor dar. Dann ist die Verdrahtung einfach $P_{kj} \leftrightarrow Q_i$. Das heißt, man verbinde die j -te Leitung des P-Busses am Prozessor k mit der k -ten Leitung des Q-Busses am Prozessor j . Diese Regel wird auf die Verwendung von Teilgruppen von Signalleitungen für Systeme erweitert, bei denen $M \neq N$.

Wie bei der Besprechung des Schaltbusses erwähnt, unterstützt der Schaltbus drei Grundarten von Datenflußmustern und Kombinationen dieser Muster. Das "Rundsende"-Datenflußmuster erlaubt einem beliebigen Prozessor die Übertragung von Daten an alle anderen Prozessoren in einem einzigen Buszyklus. Dieses Flußmuster gleicht der bei herkömmlichen Tristate-Busstrukturen angetroffenen Fähigkeit. Der Rundsendedatenfluß wird zur Verteilung von Konfigurationsdaten in einem RSP-Bündel benutzt. Der PN-Prozessor benutzt auch das Rundsenden zur Einleitung von spezifischer Verarbeitung in den Datenprozessoren durch Direktausführung von Befehlen. Im MIND-Berechnungsstil leiten diese Befehle im wesentlichen die Ausführung von funktionsmäßig abhängigen Teilprogrammen ein. Im SIND-Berechnungsstil sind die Befehle typischerweise "Funktionsanforderungs"-Befehle für die konfigurierbare Logikanordnung.

Der Schaltbus unterstützt auch allgemeine "Datenrotations"-Datenflußmuster auf Bitebene und Wortebene. Diese werden sowohl für SIND- als auch MIMD-Parallelverarbeitungsansätze für Prozessor-Prozessor-Datenfluß benutzt. Die Rotationsdatenflußmuster können als eine Erweiterung der "Ring-" Zusammenschaltung betrachtet werden, die erfolgreich in vorherigen Systemen benutzt worden ist.

Die dritte Hauptart von Datenfluß über den Schaltbus wird "Datenreflexion" genannt. Dies ist das allgemeinste und leistungsfähigste Datenflußmuster und erlaubt die Implementierung sehr komplexer Datenflußoperationen wie beispielsweise den im nächsten Abschnitt beschriebenen multidimensionalen Speicherzugriff. Reflexionsdatenfluß basiert auf dem in den meisten Logikauslegungs- und Codiertheorie-büchern beschriebenen Prinzip des "Hamming-Abstands". Beispielsweise können Daten zwischen zwei Prozessoren j und k durch Einstellen des Schaltbusreflexionsabstandes auf $d = j \otimes k$, wobei " \otimes " das bitweise exklusive ODER der Bit der Ganzzahlwerte von j und k ist, in einem Buszyklus ausgetauscht werden. Der auf diese Weise berechnete Abstand zwischen Prozessoren wird der Hamming-Abstand genannt. Die auf dem Hamming-Abstand basierenden Datenflußmuster sind ein integraler Aspekt komplexer Algorithmen wie beispielsweise der schnellen Fourier-Transformation (FFT-Fast Fourier-Transform), die für die beabsichtigten Anwendungen der RSP-Architektur von grundlegender Bedeutung sind.

Wie bemerkt, sind in der Tabelle 8 (Schaltbus-Datenflußoperationen) die Datenflußoperationen zusammengefaßt, die durch den PN-Prozessor eingeleitet werden können. Die Spalten "Buscode" und "Abstand" sind Felder in dem vom PN-Prozessor ausgegebenen Vektor "C". Jeder RSP-Prozessor bestimmt die für einen bestimmten Buszyklus durchzuführende Operation unter Verwendung des Buscodewertes. Zusätzlich zu den Datenflußoperationen Rundsenden, Rotation und Reflexion werden Buszyklen für Konfigurationsdaten, Befehlsausführung und "Assoziativteilmengen"-Verarbeitung durchgeführt. Die Assoziativverarbeitungsoperationen erlauben der PN-Verarbeitung, eine Teilmenge der Pi-Prozessoren auszuwählen, die auf nachfolgende Buszyklen reagieren wird. Dies wird beispielsweise zur Auswahl einer SIND-Funktionsteilmenge von Prozessoren für einen spezifischen Algorithmus, der Teil einer im allgemeinen funktionsmäßig partitionierten MIMD-Systemstruktur ist, benutzt.

Wenn die Anzahl der Prozessoren N größer als die Anzahl von Bit M in den Worten ist, benutzt der PN-Prozessor die Buszyklustypen "Gruppenauswahl" zur Auswahl von Teilmengen der Prozessoren zur Aktivierung. Bei diesen größeren Systemen wird für jeden Buszyklus ein zusätzlicher Taktzyklus benötigt, um volle "Breitwort"-Datenflußmuster zu vervollständigen.

Anwendungsbeispiele

Die ALP-Mittel im RSP können auf drei unterschiedliche Weisen benutzt werden: E/A-Pipeline. Schnittstelle zu externen Sensoren und Betätigungsgliedern unter Verwendung der konfigurierbaren E/A-Anschlüsse (wie beispielsweise die Datenmengen A und B). Damit verringert sich die Anzahl externer Bauelemente, die zur Implementierung eines Systems benötigt werden. In dieser Rolle wird der ALP als herkömmliche FPGA-Struktur benutzt, um mit den Steuersignalen der externen Vorrichtungen auf flexible Weise in Wechselwirkung zu treten. Das PDP-Programm greift dann auf die externen Daten unter Verwendung von Pipelinebuslese- und -schreibbefehlen zu. Der ALP kann Vor- und Nachverarbeitungsfunktionen, wie beispielsweise Bitzusammenfügung und -neuzusammenfügung bieten.

Programmierte Pipeline. Der ALP wird durch Durchführung von einfachen Overheadfunktionen für die Berechnung, die sonst mehrfache Befehle erfordern, zur Beschleunigung des Berechnungsvorgangs benutzt.

Selbständige Pipeline. Die gesamte Berechnung oder ein Großteil derselben wird im ALP durchgeführt, der direkt ohne Programmsteuerung auf die Speicher- und Schaltbus-E/A-Mittel zugreift.

Diese Ansätze können auf verschiedene Weisen kombiniert werden, um mit minimalen externen Bauelementen eine hohe Leistung bereitzustellen. In den folgenden Abschnitten wird beschrieben, wie sich diese Verfahren auf bekannte Berechnungen anwenden lassen.

Gleitkommaverarbeitung

Wenn der PDP-Prozessor im RSP keine Gleitkomma-Arithmetikfähigkeit besitzt, kann der ALP durch Durchführung der Exponentenberechnungen, Ausnahmezustanderkennung und Mantissenwortbildung zur Beschleunigung einer Menge von Gleitkomma-Unterprogrammen um einen Faktor 5 benutzt werden. Die PDP-Befehle werden dann zur Durchführung der eigentlichen Multiplizier-, Addier- und Subtrahieroperationen benutzt. Für die gegenwärtige Besprechung wird das in der Fig. 23 gezeigte 32-Bit-IEEE-Gleitkomma-Datenformat benutzt. Jedes Gleitkommawort besteht aus zwei 16-Bit-Worten. Das niederwertige Wort (ML) stellt die niederwertigen 16 Bit der Mantisse dar. Die niederwertigen 7 Bit des höherwertigen Wortes (MH) sind die höherwertigen Bit der Mantisse, das (die) höchstwertige(n) Bit des höherwertigen Wortes sind das Zeichenbit und die übrigen Bit im höherwertigen Wort (E) stellen den Exponenten dar. Im IEEE-Format wird ein "verstecktes Bit" in der Mantisse angenommen. Für alle Werte außer 0 muß an das ME-Feld ein Einzelbit mit einem Wert "1" angehängt werden, um einen richtigen 24-Bit-Mantissenwert zu erhalten.

Fig. 24 ist ein Blockschaltbild für eine ALP-Schaltung zur Durchführung von beschleunigten Gleitkomma-Operationen. Der allgemeine Plan für die ALP-Beschleunigerschaltung besteht darin, zwei Worte, die einen Gleitkomma-Wert darstellen, unter Verwendung des DRm-Pipelinebusses aus dem Speicher auszulesen. Die ALP-Schaltung bestimmt dann, ob mit dem Wert irgendwelche Ausnahmebedingungen verbunden sind. Für die gegenwärtige Besprechung wird eine verringerte Menge von Ausnahmen benutzt, die für Digitalsignalverarbeitungsanwendungen typisch sind. Diese Bedingungen sind in Tabelle 13 zusammengefaßt.

Tabelle 13

IEEE-Gleitkomma-Ausnahmen für DSP

E-Wert	M-Wert	Name	Bedingung
0	0	Null	Wert ist 0
0	0	kN	Ungültiger Wert
>0, <255	beliebig	OK	Gewöhnlicher Wert
255	beliebig	kN	Ungültiger Wert

Wenn Daten aus dem Speicher zur ALP-Schaltung ausgelesen werden, werden die Werteregister (VL und VH) mit dem Eingangswert beladen. Die Schaltungen "M=0" und "E=0" berechnen dann die Eigenschaften des Eingangswertes. Die Werteschaltung (VC) faßt dann diese Bedingungen zusammen und es wird, falls der Zustand "kN" (keine Nummer) erkannt wird, das DRm-Pipeline-Statusbit (das Signal DRm_STAT) gesetzt. Damit kann die Softwarefolge ungültige Eingangsdatenstände erkennen. Nach Abschluß einer Gleitkomma-Operation hält das Gleitkomma-Akkumulatorregister (AL und AH in der Figur) den Ergebniswert. Diese selben Zustände werden dann für den Ergebniswert erkannt und bei Erkennung eines ungültigen Ergebnisses wird das Statusbit-signal DWm_STAT gesetzt. Der sich ergebende ungültige Zustand wird ebenfalls gesetzt, wenn ein Überlauf- oder Unterlaufzustand erkannt wird.

Die ALP-Schaltung der Fig. 24 enthält einen Prozessor vollständiger Exponenten (EP) und einen Prozessor vollständiger Vorzeichenbit (SBP). Für diese Funktionen werden keine PDP-Befehle benötigt. Nunmehr werden die Funktionsweise der ALP-Schaltung und die PDP-Programmfolgen zur Durchführung der Grund-Gleitkomma-Funktionen beschrieben.

Beladen Gleitkomma-Akkumulator — Zum Beladen des Gleitkomma-Akkumulators belädt das Programm zuerst die Eingangswerteregister mit zwei Worten und setzt das Funktionsregister (F) in der ALP-Schaltung auf eine Operation "Laden". Die Gleitkomma-Steuerungs-(FPC-floating point control)Schaltung überträgt dann die VL- und VH-Registerwerte zu den AL- und AH-Registern und setzt das Signal DWm_STAT, wenn der Wert ungültig ist. Die grundlegenden Programmschritte in Assemblersprache sind folgende:

LDALPDA;	Laden niederwertiges Wort in ALP-Schaltung
LDALPDA+1;	Laden hochwertiges Wort in ALP-Schaltung
LDALPF "LDA";	Setzen ALP-Funktion
IFC "ALP_DW"...	Prüfen auf gültigen Operandenwert

Speichern Gleitkomma-Akkumulator — Das Programm setzt das Funktionsregister (F) in der ALP-Schaltung auf eine Operation "Speichern". Die Gleitkomma-Steuerungs-(FPC-)Schaltung überträgt dann die AL- und AH-Registerwerte zum DWM-Ausgangspipelinebus zur Speicherung im Speicher. Die grundlegenden Programmschritte in Assemblersprache sind die folgenden:

5 LDALPF "LDA"; Setzen ALP-Funktion auf Speichern
 STALPD A; Speichern niederwertiges Wort aus ALP-Schaltung in "A"
 STALPD A + 1; Speichern hochwertiges Wort aus ALP-Schaltung in "A + 1"

10 Gleitkomma-Vergleich — Das Programm belädt das Eingangsregister mit dem mit dem aktuellen Gleitkomma-Akkumulatorwert zu vergleichenden Wert und setzt dann das Funktionsregister auf "Vergleich". Das Signal DRm_STAT zeigt eine ungültige Eingabe an, DWM_STAT zeigt an, daß "A = V", und DRp_STAT zeigt an, daß "A > V".

15 LDALPD A; Laden niederwertiges Wort in ALP-Schaltung
 LDALPD A + 1; Laden hochwertiges Wort in ALP-Schaltung
 LDALPF "CMP"; Setzen ALP-Funktion
 IFC "ALP_DR"...; Prüfen auf aktuell größer als Eingabe

20 Gleitkomma-Multiplikation — Das Programm belädt das Eingangsregister mit dem mit dem aktuellen Gleitkomma-Akkumulatorwert zu multiplizierenden Wert und setzt dann das Funktionsregister auf "Multiplikation". Das Signal DRm_STAT zeigt eine ungültige Eingabe an. Die ALP-Schaltung berechnet automatisch den zeitweiligen Wert des sich ergebenden Exponenten und setzt das Signal DRp_STAT, wenn der Eingangswert "0" beträgt. Dann werden drei Multiplikationsoperationen unter Verwendung von Werten durchgeführt, die aus der DRp-Pipeline in einer festen Folge ausgelesen werden, die durch die Steuerschaltung definiert wird, die die Multiplexer zur Bereitstellung der Operanden in der ordnungsgemäßen Reihenfolge steuert. Das Ergebnis wird dann aus der WDP-Pipeline vom PDP aus in den Akkumulator eingeladen.

30 FPMUL:
 LDALPD A; Laden niederwertiges Wort in ALP-Schaltung
 LDALPD A + 1; Laden hochwertiges Wort in ALP-Schaltung
 LDALPF "MUL"; Setzen ALP-Funktion
 IFC "ALP_DR"...; wenn Eingangsoperand 0 ist, zu Ausgabe umspeichern
 CLR; Löschen PDP-Akkumulator
 LDT "ALPD"; Beladen PDP-Multiplikandregister mit AL
 MAC "ALPD"; Multiplizieren AL mit VH
 LDT "ALPD"; Beladen PDP-Multiplikandregister mit AH
 MAC "ALPD"; Multiplizieren AH mal VL mit Akkumulieren
 SHF 12; rechts verschieben Akkumulator
 MAC "ALPD"; Multiplizieren Akkumulation Ah mal VH
 STL "ALPV"; Übertragen niederwertige Bit zu ALP
 STH "ALPV"; Übertragen höherwertige Bit zu ALP
 IFC "DWM"...; Überprüfen auf unzulässiges Ergebnis

50 In diesem Codebeispiel stellen die Adressen "ALPD" und "ALPV" konfigurierbare Adreßmodusreferenzen zur Übertragung der ALP-WDa-Pipeline zum PDP bzw. Speicher dar. Dies ist ein Beispiel des wirklichen Vermögens der Verwendung von ALP-Pipelineschaltungen. Durch die Verwendung von einfachen Schaltungen im ALP zur Durchführung der einfachen Bithandhabungs- und Exponentenverarbeitungsfunktionen reduziert sich die Programmfolge auf ein Fünftel der Anzahl von Befehlen, die sonst zur Durchführung der IEEE-Gleitkomma-Multiplikation benötigt werden würden.

55 Gleitkomma-Addition — Das Programm belädt das Eingangsregister mit dem zum aktuellen Gleitkomma-Akkumulatorwert hinzuzufügenden Wert und setzt dann das Funktionsregister auf "Addition". Das Signal DRm_STAT zeigt eine ungültige Eingabe an. Der ALP-Exponentenprozessor berechnet den Mantissenverschiebungswert, der zum Skalieren des kleineren Operanden benutzt wird, und setzt das Signal DRp_STAT, wenn die Operation in Wirklichkeit statt dessen eine Subtraktion erfordert. Die ALP-Schaltung erkennt, welche
 60 Mantisse größer ist, und lädt diesen Wert zuerst in den PDP. Das Programm führt dann die Addition oder Subtraktion durch. Es wird eine Prüfung durchgeführt, ob Nachnormierung erforderlich ist (die Einzelheiten sind der Kürze halber weggelassen). Das Ergebnis wird dann zusammen mit dem vom Exponentenprozessor berechneten neuen Exponentenwert zurück in den Gleitkomma-Akkumulator umgespeichert.

65

ADD:		
LDALPD A;	Laden niederwertiges Wort in ALP-Schaltung	
LDALPD A + 1;	Laden hochwertiges Wort in ALP-Schaltung	
LDALPF "ADD";	Setzen ALP-Funktion auf Addition	5
LDA "ALPD";	Laden niederwertiger Teil des Augenden Mantisse	
LDT "ALPD";	Laden höherwertiger Teil des Augenden Mantisse	
LDS "ALPD";	Laden Verschiebungszählung	
IFC "ALP DR" ADD1;	wenn Subtraktion benötigt, gehe zu Punkt ADD1	
ADS "ALPD";	Addieren niederwertigen Teil des Addenden verschoben nach rechts	10
ADHS "ALPD";	Addieren höherwertigen Teil des Addenden verschoben nach rechts	
JMP ADD2;	Springen zum Ausgabesicherungsschritt	
ADD1:		
SUBS "ALPD";	Subtrahieren verschobenes Niederwertiges	15
SUHS "ALPD";	Subtrahieren verschobenes Höherwertiges	
ADD2:		
IF "upper" ...	Prüfen auf Nachnormierung erfordert	
STL "ALPV";	Speichern niederwertige Mantisse in ALP	
STH "ALPV";	Speichern höherwertige Mantisse in ALP	20

Eine ähnliche Folge wird zur Durchführung von Gleitkomma-Subtraktion benutzt. Der einzige Unterschied besteht in der Umkehrung der Richtung der Subtraktionsbedingung.

Berechnung der schnellen Fouriertransformation (FFT-Fast Fourier Transform) 25

Bei der Auswertung und Handhabung von Daten ist die Benutzung der diskreten Fouriertransformation (DFT - Discrete Fourier Transform) weit verbreitet. Die mathematische Definition des Problems ist folgende: Bei einem gegebenen Vektor von Daten X mit N Punkten ist die diskrete Fouriertransformation der N-Punktvektor y, gegeben durch das Produkt von X und der Transformationsmatrix M: 30

$$Y = M \cdot X$$

Im allgemeinen sind die Elemente von X komplexe Zahlen, die die Werte einer Zeitbereichsfunktion darstellen, und die Elemente von Y sind komplexe Zahlen, die die Koeffizienten des Frequenzbereichsspektrums der Funktion darstellen. Die Elemente der Transformationsmatrix M sind im allgemeinen komplexe Zahlen, die durch folgende Formel gegeben sind: 35

$$M_{jk} = \cos(2\pi jk/N) - i \cdot \sin(2\pi jk/N), j = 0, 1, \dots, N-1 \text{ und } k = 0, 1, \dots, N-1 \quad 40$$

wobei j und k die Zeilen- und Spaltenindizes und $i = \sqrt{-1}$ sind. Bei direkter Berechnung erfordert die DFT N^2 -Multiplizier- und N^2 -Addieroperationen. Die Algorithmen der schnellen Fouriertransformation (FFT) nutzen die Symmetrieeigenschaften der Koeffizienten M_{jk} , um die Anzahl von Berechnungen bedeutsam zu verringern. 45

Fig. 25 zeigt den Datenfluß und die Berechnungen des grundlegenden FFT-Algorithmus für $N = 8$. Wenn N eine Potenz von 2 ist, so daß $N = 2^m$, erfordert der Vorgang m + 1 Stufen. In der ersten Stufe (Stufe 0 in der Fig. 24) wird der Eingangsdatenvektor X durch die "Bitumkehr"-Permutation permutiert, um den Zwischendatenvektor $Z[j,0]$ zu bilden. In dieser Stufe wird jedes Element $X[n]$ des Vektors X so auf das Element $Z[j,0]$ abgebildet, daß die Bit mit Index j den Kehrwert des Indexes n bilden. In den Berechnungsstufen wird jedes "Reflexionspaar" von Zwischenwerten $Z[j,k]$ und $Z[j^{\wedge}k]$ transformiert, um das nächste Paar von Zwischenwerten $Z[j,k+1]$ und $Z[j^{\wedge}d,k+1]$ zu bilden. Für die Stufe k wird der Abstand zwischen den Elementen eines Paares durch die Eamming-Abstandsfunktion gegeben: 50

j ist mit $j^{\wedge}2^k$ gepaart, 55

wobei " \wedge " das bitweise exklusive ODER der binären Darstellung der Ganzzahlen j und 2^k ist. Die reduzierte Menge von Multiplikationskoeffizienten in der Figur basiert auf den Symmetrieeigenschaften der M-Matrix. insbesondere ist: 60

$$C[r] = M_{1,r} \quad 60$$

wobei der Index r für den Koeffizienten der j-ten Zeile der Stufe k durch die niederwertigen (m-1) Bit von j gegeben ist, wobei die niederwertigen (m-k) Bit auf Null gesetzt sind.

Nunmehr wird eine spezifische Ausführungsform der FFT unter Annahme der Verwendung von 16 RSP-Bau- 65
elementen, die unter Verwendung des Schaltbusses und einem 17-ten RSP, der zur Steuerung der Verarbeitungsanordnung benutzt wird, miteinander verbunden sind, dargestellt. Jeder RSP in dieser Struktur wird eindeutig durch den Prozessorindexwert L identifiziert, wobei $L = 0, 1, \dots, 15$ die "Datenprozessoren" sind und $L = 16$ die Berechnungssteuerung darstellt. Für Darstellungszwecke wird ein Wert von $N = 1024$ benutzt. Auch wird

angenommen daß der Prozessor P0 die X-Elemente $X[0], X[1], \dots, X[63]$ enthält, der Prozessor P1 die X-Elemente $X[64], X[65], \dots, X[127]$ enthält und so weiter.

In der ersten Stufe muß der Vektor X bitweise rückwärts permutiert werden. Dies läßt sich durch Anwendung von einfachen Pipelines im ALP durchführen. In jedem Schritt werden in jedem Schaltbuszyklus 16 Datenpaare ausgetauscht. Der Algorithmus für den Vorgang basiert auf der Bildung von drei Gruppen für die Bit des Indexes des Datenelements $X[i]$, das mit dem Datenelement $X[j]$ so ausgetauscht werden muß, daß j der Wert von i mit umgekehrten Bit ist. Bei $N = 1024$ werden 64 Zyklen zur Vollendung der Umkehrpermutation benötigt (16 stellt das 6-te Bit von Index I dar und so weiter):

(19,18,17,16) (15,14) (13,12,11,10) → (10,11,12,13) 14,15 (16,17,18,19)
 SP SG SV → DP DG DV

Bei der gegebenen angenommenen Datenspeicherordnung stellen die oberen 4 Bit (das mit SP bezeichnete Feld) die Prozessorkennnummer dar und die unteren 6 Bit des Indexes (SG, SV) stellen den Index für die Speicherung von Datenelementen in diesem Prozessor dar. Zwei Datenelemente $X[i]$ und $X[j]$ können unter Verwendung von Schaltbusreflexionen ausgetauscht werden, wenn der Hamming-Abstand zwischen den Ursprungs- und Zielprozessornummern einer der zulässigen Reflexionswerte d ist, wobei $d = 0, 1, \dots, 15$. Das kann durch folgendes Erfordernis erzwungen werden:

$$DP = SV \wedge d$$

Alle 1024 Werte werden ordnungsgemäß durch Durchlaufen aller möglichen Werte von SG und SV permutiert. Beispielsweise wird der Austausch von $X[153]$ gegen $X[356]$ folgendermaßen durchgeführt:

(0,0,1,0) (0,1) (1,0,1,0) ↔ (0,1,0,1) (1,0) (0,1,0,0)
 SP SG SV ↔ DP DG DV

wobei der Schaltbusabstand $d = DP \wedge SV = 14$. Bei diesem Beispiel führen die ALP-Pipelineschaltungen die detaillierte Bithandhabung durch, um das Quelladressen-(RA-Pipeline), Zieladressen- (WA-Pipeline), Quelldatenlese-(RD-Pipeline), Quellschreib-(WD-Pipeline), Schaltbussende-(XW-Pipeline) und Schaltbusempfangs-(XR-Pipeline) Signal zu erstellen, womit der größte Teil der Pipelinebusanordnungsmittel genutzt wird. In dieser Stufe der FFT wird der PDP im Prozessor P16 nur für die Folgesteuerung durch die zur Vollendung des Vorgangs benötigten 64 Schritte benutzt.

Die Fig. 26 ist ein Blockschaltbild für eine ALP-Schaltung, die in jedem Datenverarbeitungselement zur Implementierung der bei der FFT benutzten Permutationsoption zur Anwendung kommt. Zwischen den Prozessoren wird die Operation durch die Schaltbuszyklen synchronisiert. In der vorliegenden Figur ist der Block mit der Bezeichnung "A" ein Register, das mit der Basisadresse im Speicher des ersten Elements im X-Datenvektor beladen ist. Der Block "GS" ist ein Sechs-Bit-Zähler, der zum Fortschalten durch die Daten benutzt wird. Die niederwertigen zwei Bit dieses Zählers sind die "Gruppen"-Nummer und entsprechen dem SG-Feld in den obigen Beispielen. Die Quelladresse (RA) bei jedem Schritt wird dadurch erstellt, daß zuerst die exklusive ODER-Verknüpfung des Stufenwertes S mit der Prozessornummer L durchgeführt wird. Dieser Wert wird mit dem Gruppenwert G verkettet, umgekehrt (was nur Drahtverbindungen erfordert) und zu der Basisadresse hinzugefügt. Die Zieladresse (WA-Pipeline) wird durch Addieren des umgekehrten Wertes von L, verkettet mit G, zu der Basisadresse gebildet. Nach der Initialisierung arbeitet die Steuerschaltung für diesen Block als selbständige Pipeline und wiederholt den Zyklus der folgenden Schritte:

- 1) Erzeugen Leseadresse in RA, Warten auf RD-Daten bereit. Auch Erstellen des später zu benutzenden Schreibadreßwertes.
- 2) Lesen Daten aus RD und Übertragen Daten in Pipeline XW über den Schaltbus. Warten auf Empfangsdaten vom Schaltbus.
- 3) Schreiben der Empfangsdaten von XR in den Speicher an der durch den WA-Wert definierten Adresse. Erhöhen des GS-Zählers.

Diese Schleife wird solange wiederholt, bis der GS-Zähler überläuft und der Vorgang dann endet. Der Fluß des P16-Programms ist allgemein wie folgt:

```

"Setzen Basisadresse in den Datenprozessoren";/*Initi-
alisierung*/
für (D = 0; D < 16; ++D)/* je der Schaltbusabstand*/
{ "Setzen Schaltbusabstand D";
    für (G = 0; G < 4; ++G)/* Fortschreiten durch
Gruppenelemente*/
        "zyklisch wiederholen Schaltbus";}

```

Man beachte, daß der größte Teil der Arbeit für die Bitumkehrphase in der ALP-Schaltung durchgeführt wird. Die übrigen Stufen des FFT-Algorithmus benutzen die PDP-Verarbeitungsfähigkeiten in jedem Prozessor zur Durchführung der Addier-, Subtrahier- und Multiplizieroperationen. Der Algorithmus wird vereinfacht, indem die Datenprozessoren in Paare aufgeteilt werden, die die Grundberechnung durchführen, wie in Fig. 27 dargestellt. Der Hamming-Abstand zwischen Paaren von Datenelementen beträgt 0 für die ersten 6 Stufen und ist durch folgendes gegeben:

$$d = 2^{(k-6)}$$

für die letzten vier Stufen.

Fig. 28 ist ein Blockschaltbild einer ALP-Pipelineschaltung, die die Adresse für die Koeffizienten, die Adresse zum Lesen und Schreiben der Daten berechnet und die Daten über den Schaltbus für die FFT-Berechnung überträgt. In dieser Schaltung ist der Block "A" ein Register, in dem die Basisadresse im Speicher für die Daten gespeichert ist, der Block "C" ein Register, das die Basisadresse im Speicher für die Koeffizienten speichert und "S" ein Register, das den Stufenindex speichert. Der Block "Dc" ist ein Zähler, der um in "S" definierte Werte ansteigt, um Adressen für aufeinanderfolgende Datenelemente für jede Stufe zu erzeugen. Der Block "Cc" ist ein Zähler, der gleichermaßen aufeinanderfolgende Koeffizientenadressen erzeugt. Der Multiplexerblock "mux" wählt die Daten oder Koeffizientenadresse zum Auslesen aus dem Speicher aus.

Ein beispielhafter Umriß eines Algorithmus für die ungeradzahlig Datenverarbeitungselemente für die Berechnungsstufen der FFT ist wie folgt: Der folgende Umriß des Algorithmus wird dann für die geradzahlig Datenverarbeitungselemente benutzt:

```

/* Berechnungen für das ungeradzahlige Datenverarbei-
tungselement */
für (I = 0; I < 64; ++I)
{ tmp = "c" * "d"; /* Koeffizienten- und Datenadresse
vom ALP */
    "Senden tmp zum geraden Prozessorpaar";
    "Empfangen Paardaten"
    "d" = Paar - tmp;}

/* Berechnungen für das geradzahlige Datenverarbei-
tungselement */
für (I = 0; I < 64; ++I)
{ tmp = "d"; /* gerade Datenadresse von ALP */
    "Senden tmp zum ungeraden Prozessorpaar";
    "Empfangen Paardaten"
    "d" = Paar + tmp;}

```

Danach wird der folgende Algorithmus vom Steuerprozessor (P16) zum Steuern des Datenflusses durch den Schaltbus benutzt:

```

/* PN FFT Steueralgorithmus */
für (Stufe = 0; Stufe < 10; ++Stufe)/* jeder Schaltbus-
    "Setzen Schaltbusabstand auf 0";
sonst
    "Setzen Schaltbusabstand auf (Stufe - 6)";
    "Rundsenden Stufenwert zu Datenprozessor";
für (I = 0; I < 64; ++I) "Zyklisch wiederholen
    Schaltbus";}

```

Bildverarbeitung

Eine gebräuchliche Aufgabe bei Videokonferenz-, Bildtelefon- und Multimedia-Computeranwendungen ist die Kompression von Videobildströmen. Der Grundansatz besteht im Vergleichen der Bilddaten aus einem Vollbild mit einem aktuellen Schätzbild, Ableiten der wesentlichen Differenz und nachfolgendem Übertragen einer codierten Version der Differenzen. Dieser Vorgang wird "Videocodierer" genannt. Danach wird ein Videodecoder zur Rekonstruktion des ursprünglichen Bildstroms durch Umkehren der Grundschritte benutzt. Bei den meisten heutigen Ansätzen wird eine hierarchische Darstellung des im Speicher gespeicherten Bildes in der in Fig. 29 dargestellten Form benutzt.

In dieser Darstellung werden die Bilddaten zuerst in Blöcke von Bildpunkten (Pel) eingeteilt, wobei jeder Block aus einem $N \times N$ -Teilbild besteht, wobei N die Anzahl von Prozessoren ist. In dem in Fig. 29 gezeigten Beispiel ist jeder Block ein 8×8 -Bild. Vier Blöcke werden zusammengruppiert, um einen "Makroblock" von 16×16 Bildpunkten zu bilden. In einem Parallelverarbeitungssystem mit mehrdimensionalem Speicherzugriff kann auf jede Zeile oder Spalte eines Bildblocks unter Verwendung der Reflexionsdatenflußmuster in einem einzigen Schaltbuszyklus zugegriffen werden.

Der vereinfachte Fluß eines Videocodieralgorithmus ist wie folgt:

- 1) Vergleichen jedes neuen Makroblocks mit dem entsprechenden Makroblock des gegenwärtigen Schätzbildes. Wenn die Differenz unterhalb eines vorgeschriebenen Schwellwertes liegt, wird keine weitere Verarbeitung an dem Makroblock durchgeführt.
- 2) Bewegungserkennung — Die meisten Differenzen in einem Makroblock sind das Ergebnis der Bewegung eines Teils des Bildes. Der Algorithmus bestimmt die Bewegungsgröße eines Makroblocks durch eine komplexe Vergleichsfolge mit dem Vergleichen einer verschobenen Version des geschätzten Makroblocks mit einer Nachbarumgebung des neuen Bildes. Daraus ergibt sich eine annähernde optimale neue Position für den Schätzblock.
- 3) Transformation — Die Differenz zwischen dem Schätzblock und dem neuen Block wird unter Verwendung des Algorithmus der diskreten Cosinustransformation (DCT-Discrete Cosine Transform) transformiert.
- 4) Der transformierte Differenzblock wird quantisiert, um die Gesamtzahl benötigter Bit zu verringern.
- 5) Die quantisierten Daten werden dann unter Verwendung von Datenkompressionsverfahren wie beispielsweise Codierung mit veränderlicher Wortlänge (VLC-Variable Length Coding) weiter reduziert.

In der folgenden Übersicht wird die Verwendung eines 17-Prozessor-RSP-Systems zur Implementierung der Bewegungserkennungs- und Transformationsschritte, die typischerweise die kompliziertesten Aspekte der Videokompression darstellen, beschrieben. Der Ansatz bedient sich der Fähigkeiten des mehrdimensionalen Zugriffs des Schaltbusses in Kombination mit der Verwendung des ALPs zur Adreßerzeugung und zum Bildpunktvergleich.

Fig. 30 zeigt die Abbildung von Bildpunkten in einem Makroblock auf die 16 RSP-Datenprozessoren, die unter Verwendung von Hexadezimaldarstellung ("0", "1", ..., "A", ..., "F") numeriert sind, für eine auf der Architektur der vorliegenden Erfindung basierende Bildverarbeitungsanwendung. Die Bildpunkte des Makroblocks werden entsprechend den auf dem Hamming Abstand-Konzept basierenden mathematischen Zuordnungsregeln den Speichern der verschiedenen RSP-Prozessoren zugewiesen. Unter diesen Regeln wird der Bildpunkt an Stelle x und y im Makroblock auf die Koordinaten P und A abgebildet, wobei P die Prozessornummer und A der Adressenversatz des Bildpunktes in der Blockspeicherung im Prozessor P ist. Die mathematische Regel ist folgende:

$$P = x \wedge R_p(y)$$

$$A = y$$

wobei $R_p(\cdot)$ der Bitumkehroperator ist und " \wedge " ein bitweises exklusives ODER anzeigt. Durch diese Regeln wird dann die umgekehrte Abbildung von (A, P) auf Koordinaten (x, y) durch folgende Gleichungen gegeben:

$$\begin{aligned} x &= P \wedge R_p(A) \\ y &= A \end{aligned}$$

5

Durch diese verwürfelte Anordnung von Daten unter den Prozessoren wird der Zugriff auf die Daten unter Verwendung mehrerer "Adressiermodi", wie in Fig. 31 dargestellt, ermöglicht. Der Grundadressiermodus ist "CM40", der erlaubt, daß jede 16-Pelpunktzeile von einem Block in einem Schaltbuszyklus gelesen werden kann. Bei diesem Modus sind die Werte y und A die ausgewählte Zeile und haben denselben Wert für alle Prozessoren. Der Schaltbus liest oder schreibt diese Zeile unter Verwendung eines Reflektionsabstandes von $R_p(y)$, der die Daten für eine Schreiboperation in die richtigen Prozessoren verwürfelt bzw. die Daten aus den Prozessoren für eine Leseoperation entwürfelt.

Als Alternative kann jede Spalte in einem Block in einem Schaltbuszyklus unter Verwendung des Adreßmodus "CM04" modifiziert werden. In diesem Modus ist der Wert x eine Konstante, der Schaltbusabstand ist auf den Wert x gesetzt und die Adresse in jedem Prozessor ist gegeben durch:

$$A = R_p(P \wedge x)$$

Die verwürfelte Abbildung erlaubt gleichermaßen den Zugriff eines beliebigen 8×2 , 4×4 - oder 2×8 -Blocks von Bildpunkten in einem Zyklus. Zusätzlich erlaubt das Schema die Verwendung von "gegabelten" Adressiermodi. Beispielsweise erlaubt der Modus "BM44" den Zugriff auf Bildpunkte in einem zweidimensionalen Muster mit einem Abstand von 4 zwischen jedem Bildpunkt.

Es gibt mehrere Weisen zur Verwendung der mehrdimensionalen Zugriffsmodi dieses Schemas zur Implementierung der Bewegungserkennungsstufe der Videocodierung. Modus BM44 kann zur Durchführung einer schnellen Überprüfung einer Bildänderung durch Vergleichen einer Teilmenge des neuen Bildblocks mit dem gegenwärtigen Schätzblock in drei Schaltbuszyklen benutzt werden. In diesem Fall wird eine ALP-Pipeline-schaltung zur Erzeugung der Speicheradressen "A", zum Durchführen eines Schwellwertvergleichs der beiden Bildpunkte und Melden des Gesamtergebnisses ("ja" bzw. "nein") an den PN-Prozessor benutzt. Wenn alle Prozessoren "keine bedeutende Differenz" melden, dann überspringt der PN-Prozessor alle weiteren Arbeiten an diesem Blockpunkt.

Wird eine bedeutende Blockdifferenz erkannt, versucht der Algorithmus, durch ein Suchverfahren die optimale neue Stelle für den Block zu bestimmen. Der allgemeine Suchplan für eine Bewegungsversatzanwendung ist in Fig. 32 dargestellt. Es stehen, mehrere Strategien zum Finden des optimalen Versatzes zur Verfügung. Im äußersten Fall wird der Schätzblock mit allen möglichen Stellen in einer festen Nachbarumgebung verglichen. Dies würde für die meisten Anwendungen unzulässig viel Zeit erfordern. Eine gebräuchliche Strategie ist, ein Verfahren des "Bezwingens durch Teilen" zu benutzen. Das Folgende ist ein grober Umriss:

- 1) Die Vergleichsergebnisse der schnellen Überprüfung auf einen Nullversatz werden durch Addieren der Differenzwerte aller 16 Prozessoren angesammelt, um einen Gesamtdifferenzschätzwert zu erhalten. Dieser Reduktionsschritt erfordert 4 Schaltbuszyklen. Im ersten Zyklus wird der Betrag der Differenz in jedem Prozessor zum vergleichbaren Wert im Prozessor mit Abstand 1 hinzuaddiert. Der Schritt wird für Abstand 2, 4 und 8 wiederholt, woraus sich die in allen Prozessoren gespeicherte Sammeldifferenz ergibt. Dies stellt den aktuellen Gesamtdifferenzwert dar.
- 2) Es wird dann die Differenz zwischen dem Schätzblockversatz gefunden, die Daten von jedem der acht Nachbarn des Schätzblocks enthält. Dies wird mit einem groben Versatz von 4 Bildpunkten durchgeführt. Für jede der acht möglichen Bewegungsrichtungen wird eine neue Sammeldifferenz berechnet. Wenn die neue Sammeldifferenz weniger als der vorherige Differenzwert ist, ersetzt die neue Sammeldifferenz den alten Wert und der neue Versatz ersetzt den vorherigen Versatzschätzwert.
- 3) Der Schritt 2 wird mit einem Versatzabstand von 2 vom aktuellen optimalen Versatz wiederholt.
- 4) Der Schritt 2 wird mit einem Versatzabstand von 1 vom aktuellen optimalen Versatz wiederholt.

Daraus ergibt sich ein Bewegungsvektor (D_x, D_y) , der eine Annäherung an den optimalen Bewegungsversatz anzeigt. Dieser Versatz wird dann zur Berechnung eines Differenzblocks benutzt, der die Differenz von Bildpunkten im neuen Bild im Verhältnis zu dem Versatzwert des Schätzblockes enthält. Der Differenzblock wird dann unter Verwendung eines zweidimensionalen DCT-Algorithmus transformiert. Dies geschieht typischerweise an den 8×8 -Blöcken, um die Menge an notwendigen Berechnungen herabzusetzen. Die allgemeine Gleichung für den zweidimensionalen Algorithmus ist wie folgt:

$$G_{xy} = \frac{2}{N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} W_{xi} \cdot W_{yj} \cdot D_{ij}$$

wobei D_{ij} die Bildpunkt-Differenzwerte an Stelle $x = i$ und $y = j$ sind, G_{xy} die Matrix transformierter Werte ist und die Transformationskoeffizienten W_{ij} durch folgende Formel gegeben sind:

$$W_{ij} = \begin{cases} \cos(\pi / 4) & , j = 0 \\ \cos\left(\frac{\pi (2i+1)j}{2N}\right) & , j \neq 0 \end{cases}$$

Der Vorgang wird dadurch vereinfacht, daß zuerst in folgenden Gleichungen eine eindimensionale DCT an den Zeilen des Blocks und danach eine eindimensionale DCT an den Spalten des Blocks durchgeführt wird:

$$H_{x,Y} = \sum_{i=0}^{N-1} W_{Y,i} \cdot \hat{D}_{i,x}, \quad Y = 0, 1, \dots, N-1$$

$$G_{x,Y} = \frac{2}{N} \sum_{i=0}^{N-1} W_{x,i} \cdot H_{x,i}, \quad X = 0, 1, \dots, N-1$$

Für die Videokompression wird die DCT typischerweise für 8×8 -Blöcke berechnet. Die Koeffizienten W_{ij} verhalten sich nicht so gut wie die entsprechenden FFT-Koeffizienten. Diese Koeffizienten weisen eine Symmetrieeigenschaft auf, die die Reduktion auf nur $N-1$ verschiedene Koeffizienten C_j durch folgende Definition erlaubt:

$$C_j = W_{0,j}, \quad j = 1, 2, \dots, N-1$$

Die eindimensionale 8×8 -Transformation läßt sich dann durch folgendes Matrixprodukt ausdrücken:

$$\begin{bmatrix} H_0 \\ H_1 \\ H_2 \\ H_3 \\ H_4 \\ H_5 \\ H_6 \\ H_7 \end{bmatrix} = \begin{bmatrix} C_4 & C_4 & C_4 & C_4 & C_4 & C_4 & C_4 & C_4 \\ C_1 & C_3 & C_5 & C_7 & -C_7 & -C_5 & -C_3 & -C_1 \\ C_2 & C_6 & -C_6 & -C_2 & -C_2 & -C_6 & C_2 & C_2 \\ C_3 & -C_7 & -C_1 & -C_3 & C_3 & C_1 & C_7 & -C_3 \\ C_4 & -C_4 & -C_4 & C_4 & -C_4 & -C_4 & -C_4 & C_4 \\ C_5 & -C_1 & C_7 & C_3 & -C_3 & -C_7 & C_1 & -C_5 \\ C_6 & -C_2 & C_2 & -C_6 & -C_6 & C_2 & -C_2 & C_6 \\ C_7 & -C_5 & C_3 & -C_1 & C_1 & -C_3 & C_5 & -C_7 \end{bmatrix} \cdot \begin{bmatrix} D_0 \\ D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \\ D_6 \\ D_7 \end{bmatrix}$$

was 64 Multiplizier-Akkumulieroperationen erfordert. Die Anzahl von Multiplizieroperationen läßt sich durch Verwendung der Symmetrieeigenschaften der Koeffizientenmatrix und Verwendung eines durch die Summe und Differenz der Daten gebildeten Hilfsdatenvektors verringern:

$$\begin{bmatrix} H_0 \\ H_1 \\ H_2 \\ H_3 \\ H_4 \\ H_5 \\ H_6 \\ H_7 \end{bmatrix} = \begin{bmatrix} C_4 & 0 & C_4 & 0 & C_4 & 0 & C_4 & 0 \\ 0 & C_1 & 0 & C_3 & 0 & C_5 & 0 & C_7 \\ C_2 & 0 & C_6 & 0 & -C_6 & 0 & -C_2 & 0 \\ 0 & C_3 & 0 & -C_7 & 0 & -C_1 & 0 & -C_5 \\ C_4 & 0 & -C_4 & 0 & -C_4 & 0 & C_4 & 0 \\ 0 & C_5 & 0 & -C_1 & 0 & C_7 & 0 & C_3 \\ C_6 & 0 & -C_2 & 0 & C_2 & 0 & -C_6 & 0 \\ 0 & -C_5 & 0 & -C_1 & 0 & -C_3 & 0 & -C_7 \end{bmatrix} \cdot \begin{bmatrix} D_0 + D_7 \\ D_0 - D_7 \\ D_1 + D_6 \\ D_1 - D_6 \\ D_2 + D_5 \\ D_2 - D_5 \\ D_3 + D_4 \\ D_3 - D_4 \end{bmatrix}$$

Mit dieser Form wird die Berechnung auf 32 Multiplizier-Akkumulieroperationen verringert, die über die 16 Datenprozessoren verteilt sind. Zur Implementierung auf dem 16-Prozessor-RSP-System werden zwei eindimensionale Transformationen gleichzeitig berechnet. Während der Zeilentransformationsphase wird der

8x2-Adreßmodus CM31 zum gleichzeitigen Zugreifen auf zwei Zeilen benutzt. Es werden dann acht Prozessoren zum Verarbeiten einer Zeile benutzt, während die übrigen acht Prozessoren die zweite Zeile verarbeiten. In der Spaltentransformationsphase wird der Adressmodus CM13 zum Zugreifen auf zwei Spalten in einem Zyklus benutzt. Der Summen-/Differenzvektor läßt sich leicht durch Verwendung der Reflexion mit Abstand 7 auf dem Schaltbus berechnen. Es werden dann ALP-Pipelines zur Berechnung der A/P- und der Koeffizientenadressen benutzt. Es folgt der Umriss des Codes für die vollständige zweidimensionale 8x8-DCT für die geradzahlgigen Prozessoren:

```

Die ALP-Schaltung // zweidimensionale 8x8-DCT; 10
geradzahlgige Prozessoren
für (Zeile = 0; Zeile < 8; Zeile = Zeile + 2) // zwei
Zeilen berechnet parallel 15
{ "Senden D"
  "Empfangen umgekehrt Dr, 8x2-Zeilenmodus";
  D = D + Dr; // Summen-/Differenzvektor 20
  T = 0; // transformierter Wert
  für (I = 0; I < 8; I = I + 2)
  { "Senden D, 8x2-Zeilenmodus, Abstand I"; 25
    "Empfangen Di-Wert";
    T = T + Di * "C"; // Adresse des Koeffizienten C
    von ALP-Schaltung } } 30

```

Nach Abschluß der Zeilentransformation werden mit einem ähnlichen Algorithmus die Spaltentransformationen berechnet: 35

```

für (sp = 0; sp < 8; sp = sp+2) // zwei Spalten
berechnet parallel 40
{ "Senden D"
  "Empfangen umgekehrtes Dr, 2x8-Spaltenmodus";
  D = D + Dr; // Summen-/Differenzvektor 45
  T = 0; // transformierter Wert
  für (I = 0; I < 8; I = I + 2)
  { "Senden D, 2x8-Spaltenmodus, Abstand I"; 50
    "Empfangen Di-Wert";
    T = T + Di * "C"; // Adresse des Koeffizienten C
    von ALP-Schaltung } 55
T = T >>2; // geteilt durch 4 }

```

Die ALP-Schaltungen für diese Berechnungen nutzen die kombinierten Datenflußoperationen des Schaltbusses. Beispielsweise wird das i-te Datenelement im Summen-/Differenzvektor unter Verwendung eines Schaltbuszyklus gelesen, der die Bildpunktabbildentwürfungsoperation mit dem Vektorelementversatz kombiniert. Nach Abschluß der DCT werden zusätzliche ALP-Schaltungen zur Durchführung der Quantisierungs- und Lauflängencodierschritte benutzt. 60

Wie bemerkt, benutzen Bildverarbeitungsanwendungen typischerweise Komplexe Datentransformationen wie beispielsweise die diskrete Fouriertransformation (DFT), diskrete Cosinustransformation (DCT) oder diskrete Walsh-Hadamard-Transformation (DWT) zur Umwandlung des Bildes in eine Ortsfrequenzdarstellung. Dies wird beispielsweise zur Bestimmung der sich in einem Bild bewegenden Art von Objekt durch Benutzung 65

von Merkmalsanpassungsmedien benutzt. Alle diese Transformationen werden bei Implementierung unter Verwendung einer Gruppe von RSP-Vorrichtungen mit einer N-fachen Geschwindigkeit (oder schneller) durchgeführt.

Eine Implementierung der DWT-Berechnung wird hier als weiteres Beispiel, wie die konfigurierbare Logik und Schaltbusverbindungsstruktur zusammen zur Beschleunigung einer Anwendung benutzen werden können, benutzt. Für den eindimensionalen Fall werden die Eingangsdatenworte $f[i]$ für $i = 0, 1, \dots, N-1$ durch folgende Gleichung in einen N-Wort-Transformationsvektor $F[i]$ umgewandelt:

$$F_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} C_{jk} \cdot f_j$$

wobei die Koeffizienten C_{jk} die Werte $+1$ oder -1 entsprechend einem Hammingabstandverhältnis sind:

$$C_{jk} = \begin{cases} +1, & \Omega(j \otimes k) = 0 \\ -1, & \Omega(j \otimes k) = 1 \end{cases}$$

In dieser Formel ist $\Omega(j \otimes k)$ das "Gewicht" des Hammingabstands, so daß $\Omega(j \otimes k)$ den Wert 0 besitzt, wenn der Abstand $j \otimes k$ geradzahlig ist und $\Omega(j \otimes k)$ den Wert 1 besitzt, wenn der Abstand ungeradzahlig ist. Eine direkte Implementierung dieser Gleichungen erfordert $N^2/2$ Additionen und $N^2/2$ Subtraktionen. Für die gegenwärtige Besprechung ist ein "schneller" Algorithmus dargestellt, der nur $N \cdot \log_2(N)$ Additionen und Subtraktionen erfordert und sich der Schaltbusreflexionsdatenflußmuster bedient. Der Algorithmus setzt zuerst die Eingangsdatenwerte f_j in den Ausgangswert des Vektors Z_0 ein. Dann werden $\log_2(N)$ Parallelverarbeitungszyklen zum Modifizieren der Elemente von Z benutzt, damit sie der gewünschte Transformationswert werden. Die erforderliche Berechnung für jedes Element j im Schritt k ist durch folgende Formel gegeben:

$$Z_j^{k+1} = \begin{cases} Z_j^k + Z_{j',k}^k & j_k = 0 \\ Z_j^k - Z_{j',k}^k & j_k = 1 \end{cases} \text{ für } k = 0, 1, \dots, \log_2(N) - 1.$$

wobei j_k der Wert des k-ten Bits der Ganzzahl j und j' der Index des Hammingabstands-"Paares" für das Element j ist, der durch $j' = j \otimes 2^k$ gegeben ist. Fig. 33 zeigt den Datenfluß und die Berechnungen des Walsh-Edamard-Algorithmus für $N = 8$. Die "X"-Datenflüsse im Diagramm stellen die zur Implementierung der Berechnungen in jedem Schritt benötigten "Reflexions"-Datenflüsse dar. Die benötigten Hammingabstände betragen 1 für die erste Spalte, 2 in der nächsten Spalte usw. Zur Implementierung einer Gruppe von acht RSP-Prozessoren unterhält jeder RSP einen Wert des Vektors. Bei jedem Schaltbuszyklus gibt jeder Prozessor seinen aktuellen Z_j -Wert aus und empfängt seinen Hammingabstand-Paarwert. Im vorliegenden Beispiel residiert das Element Z_j im Prozessor mit der Nummer j , so daß der Prozessor eine Addition durchführt, wenn $j_k = 0$ ist, oder eine Subtraktion durchführt, wenn der Wert von $j_k = 1$ beträgt. Die Fig. 34 ist ein Blockschaltbild einer konfigurierbaren Logikschaltung, die die Berechnung der Fig. 33 implementiert. Die Schaltung benutzt vier der Leitungssätze der Pipelinebusanordnung, RD, um den Ausgangswert vom chipinternen Speicher zu holen, XW und XR für den Schaltbusdatenfluß und WD, um das Ergebnis in den lokalen Speicher zurückzuschreiben. Durch Verwendung der selbstständigen Pipelinefähigkeit kann die gesamte Berechnung ohne Ausführung irgendwelcher Skalaprozessorbefehle durchgeführt werden. Der PN-Prozessor wird zum Fortschalten durch den Algorithmus und Synchronisieren der Operationen unter Verwendung der Schaltbusdatenflußzyklen benutzt.

Dieser Grundalgorithmus läßt sich für einen beliebigen Wort von Datenelementen N erweitern und kann zur Implementierung der zweidimensionalen Walsh-Hadamard-Transformation für Bildverarbeitung erweitert werden. Für die zweidimensionale Transformation werden die Daten entsprechend den schon beschriebenen Regeln für multidimensionalen Speicherzugriff verteilt. Die Transformationsoperationen werden dann an den Spalten der Bildmatrix und danach an den Zeilen der Matrix durchgeführt. Die Endwerte werden durch eine Rechtsverschiebung von $\log_2(N)$ zur Implementierung des Teilungsfaktors $1 : N$ an der ursprünglichen Gleichung eingestellt. Die Fig. 35 zeigt einen Teil eines verschiebbaren Pipelinesegments zur Verwendung bei der Ausführung der Berechnung der Fig. 33. Diese detaillierte Schaltungssynthese und -auslegung wurde unter Verwendung eines RSP-Datenweggeneratorsoftwarewerkzeugprototyps erstellt.

Patentansprüche

1. Rekonfigurierbares Rechenbauelement mit einem adaptiven Logikprozessor, dadurch gekennzeichnet, daß der Logikprozessor eine Mehrzahl einzeln konfigurierbarer Logikzellen (150), die in einer Anordnung angeordnet sind, eine Mehrzahl von vertikalen Spalten konfigurierbarer Logikzellen (150) und eine Mehrzahl horizontaler Zeilen konfigurierbarer Logikzellen (150) enthält, umfaßt, wobei ein Satz Steuerlei-

- tungen (134) zur Übertragung von Steuersignalen und ein Satz Datenleitungen (132) zur Übertragung von Daten zwischen einem Pipelinesegment (160) und einer Logikzellenrekonfigurationssteuerung vorgesehen ist, wobei das Pipelinesegment (160) eine Menge konfigurierter Logikzellen enthält und entlang der vertikalen Erstreckung der Anordnung positioniert ist, wobei der Satz von Steuerleitungen (134) und der Satz von Datenleitungen (132) sich über eine Mehrzahl von Spalten der Anordnung erstrecken und damit erlauben, daß sich das Pipelinesegment (160) an einer Mehrzahl horizontaler Stellen der Anordnung befindet, während mit einer Logikzellenrekonfigurationssteuerung die Konfiguration der Logikzellen (150) zum Bilden des Pipelinesegments (160) und Ausführung der Menge von Logikfunktionen durch die konfigurierten Logikzellen (150) durch Übertragung von Befehlen und Daten mittels der Steuerleitungen (134) und Datenleitungen (132) steuerbar ist.
2. Rechenbauelement nach Anspruch 1, dadurch gekennzeichnet, daß eine Arithmetikverarbeitungseinheit zur Ausführung von skalaren Arithmetikfunktionen vorgesehen ist und der adaptive Logikprozessor weiterhin einen zweiten Satz von Datenleitungen zur Übertragung von Daten zwischen der im Pipelinesegment enthaltenen Menge konfigurierter Logikzellen und der Arithmetikverarbeitungseinheit umfaßt, wobei sich der zweite Satz Datenleitungen über jede Spalte der Anordnung erstreckt.
3. Rechenbauelement nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß der adaptive Logikprozessor einen zwischen benachbarten Zeilen konfigurierbarer Logikzellen verlaufenden Zeilenlokalbus, wobei die konfigurierbaren Logikzellen in besagten Nachbarzellen selektiv damit verbindbar sind, Logikzellen-Lokalbusschnittstellenschaltungen, die selektiv zwischen einer ausgewählten konfigurierbaren Logikzelle und einem ausgewählten Zeilenlokalbus verbindbar sind, um zuzulassen, daß die besagte ausgewählte konfigurierbare Logikzelle Daten aus dem ausgewählten Zeilenlokalbus ausliest oder Daten in denselben einschreibt, und Direktverbindungsmittel umfaßt, die zwischen der ausgewählten konfigurierbaren Logikzelle und einer benachbarten konfigurierbaren Logikzelle verbindbar sind, so daß ein durch die ausgewählte konfigurierbare Logikzelle bereitgestelltes Ausgangssignal unabhängig vom Zeilenlokalbus direkt als Eingangssignal für die benachbarte konfigurierbare Logikzelle bereitgestellt werden kann.
4. Rechenbauelement nach Anspruch 3, dadurch gekennzeichnet, daß der adaptive Logikprozessor einen zwischen benachbarten Spalten konfigurierbarer Logikzellen verlaufenden Spaltenlokalbus, wobei die konfigurierbaren Logikzellen in besagten Nachbarspalten selektiv damit verbindbar sind und weiterhin wobei die Logikzellen-Lokalbusschnittstellenschaltungen selektiv zwischen einer ausgewählten konfigurierbaren Logikzelle und einem ausgewählten Spaltenlokalbus verbindbar sind, um zuzulassen, daß die ausgewählte konfigurierbare Logikzelle Daten aus dem ausgewählten Spaltenlokalbus ausliest oder Daten darin einschreibt.
5. Rechenbauelement nach einem der Ansprüche 1 bis 4, dadurch gekennzeichnet, daß die zwischen der Menge konfigurierter Logikzellen und der Logikzellenrekonfigurationssteuerung übertragenen Steuersignale ein Funktionsanforderungssignal, das anzeigt, daß die Logikzellenrekonfigurationssteuerung einen Befehl in einer ausführenden Befehlsmenge identifiziert hat, der zur Ausführung durch die Menge konfigurierter Logikzellen bestimmt ist, ein Datenanforderungssignal, das anzeigt, daß die Logikzellenrekonfigurationssteuerung Daten identifiziert hat, die von einem anderen Element des rekonfigurierbaren Rechenbauelements zum adaptiven Logikprozessor oder vom adaptiven Logikprozessor zu einem anderen Element des rekonfigurierbaren Rechenbauelements zu übertragen sind, und ein selbständiges Datenübertragungssignal zur Steuerung der Ausführung einer selbständigen Datenübertragung zwischen dem adaptiven Logikprozessor und einem anderen Element des rekonfigurierbaren Rechenbauelements umfassen, wobei die selbständige Datenübertragung unabhängig von einem Befehl im ausführenden Befehlssatz ist.
6. Rechenbauelement nach einem der Ansprüche 1 bis 5, dadurch gekennzeichnet, daß eine zum adaptiven Logikprozessor externe Speichervorrichtung zur Speicherung von Logikzellenkonfigurationsdaten vorgesehen ist, wobei die Speichervorrichtung durch den zweiten Satz von Datenleitungen mit den anderen Elementen des rekonfigurierbaren Rechenbauelements verbunden ist.
7. Rechenbauelement nach Anspruch 6, dadurch gekennzeichnet, daß ein externer Adreßgenerator zum Erzeugen von zum Zugreifen auf die Speichervorrichtung benutzten Speicheradressen vorgesehen ist, wobei der Adreßgenerator durch den zweiten Satz Datenleitungen mit den anderen Elementen des rekonfigurierbaren Rechenbauelements verbunden ist.
8. Rekonfigurierbares Rechensystem mit einer Mehrzahl miteinander verbundener rekonfigurierbarer Rechenbauelemente nach einem der Ansprüche 1 bis 7 und mit einem Verbindungsbus zur Bereitstellung von Signalübertragung zwischen der Mehrzahl rekonfigurierbarer Rechenbauelemente, wobei eine Netto-Signalbandbreite der Verbindung im Verhältnis zur Anzahl miteinander verbundener rekonfigurierbarer Rechenbauelemente zunimmt.
9. Rechensystem nach Anspruch 8, dadurch gekennzeichnet, daß jedes der rekonfigurierbaren Rechenbauelemente einen Verbindungsbuskoppler zum Verbinden des rekonfigurierbaren Rechenbauelements mit dem zweiten Satz Datenleitungen und dem Verbindungsbus umfaßt.

Hierzu 23 Seite(n) Zeichnungen

Fig. 1 RSP-Verarbeitungsmodell

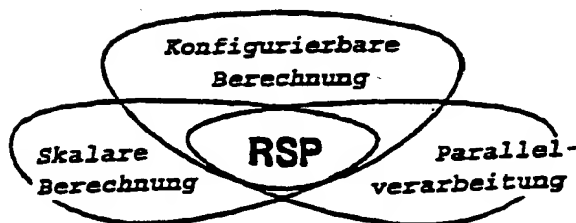


Fig. 2

Allgemeines rekonfigurierbares Signalverarbeitungssystem

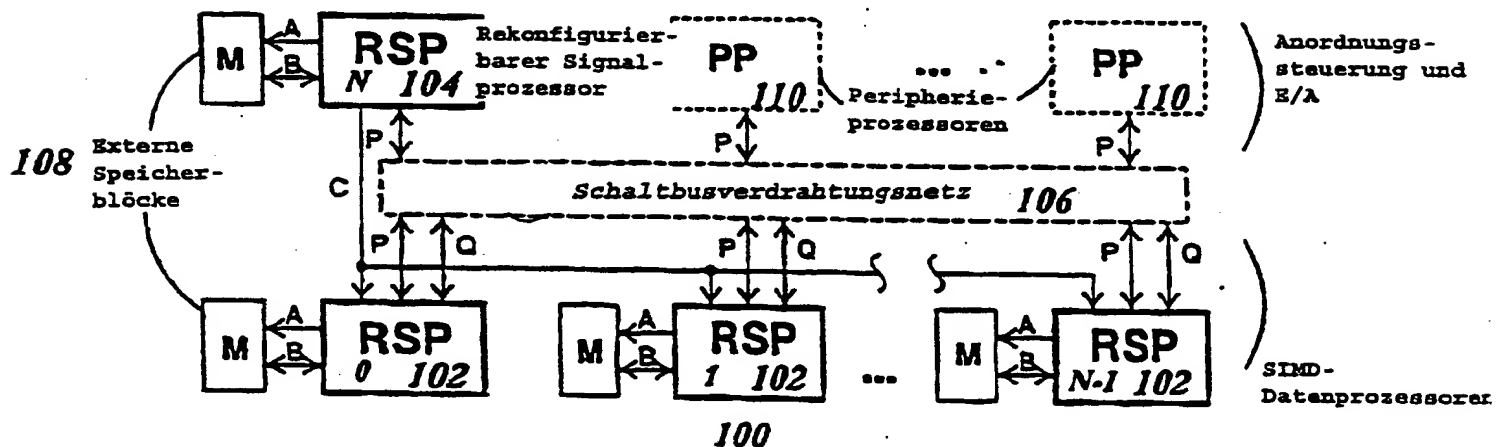


Fig. 3 RSP-Grundblöcke und Busanordnung

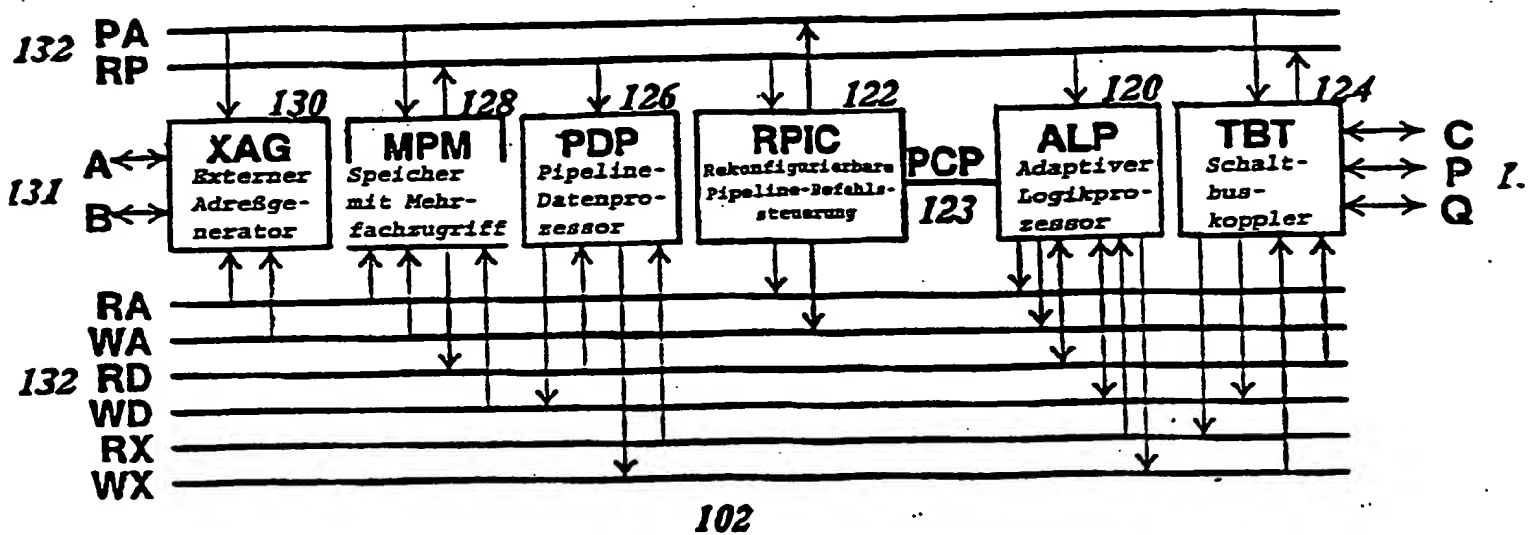


Fig. 4 RSP-Befehlsformate

Typ ZA	OP			
Typ SC	OP		C	
Typ SM	OP		0	A
Typ SM Indirekt	OP		1	AM R
Typ MS	OP	S	I	A
Typ MR	OP	R	I	A
Typ LC	OP		i	A
	C			
Typ LM	OP		i	A
	A			

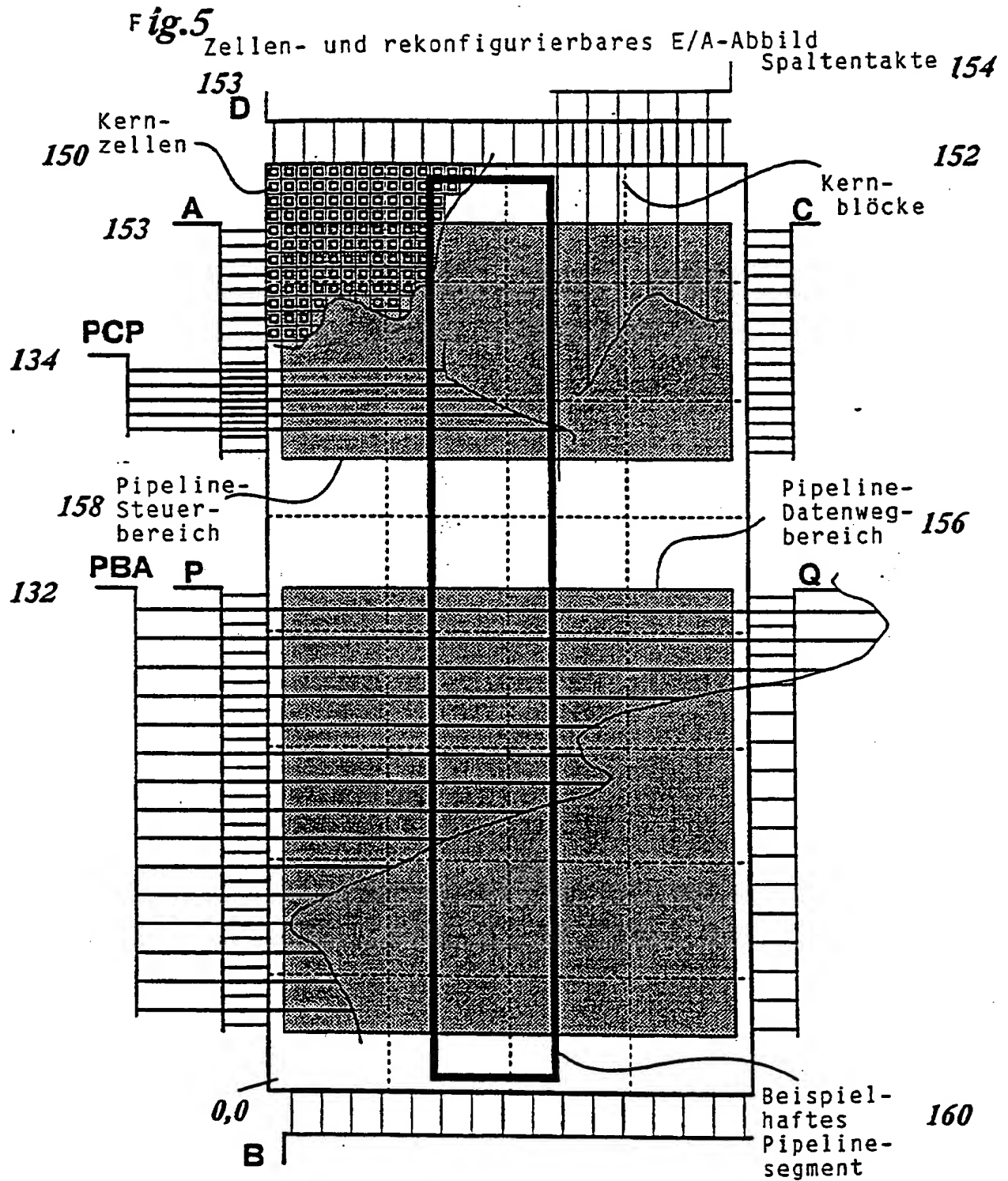


Fig. 6 PDP-Befehlspipelineausführung

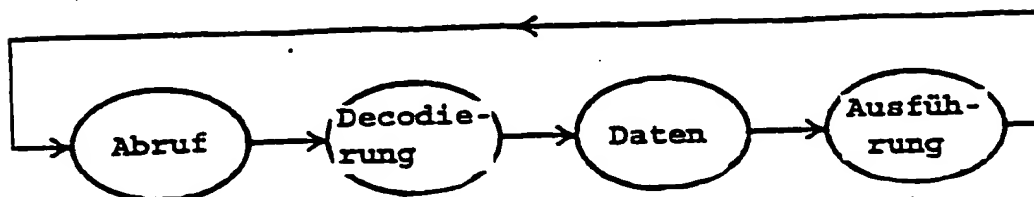


Fig. 7 ALP-Pipelinsteuerschnittstelle

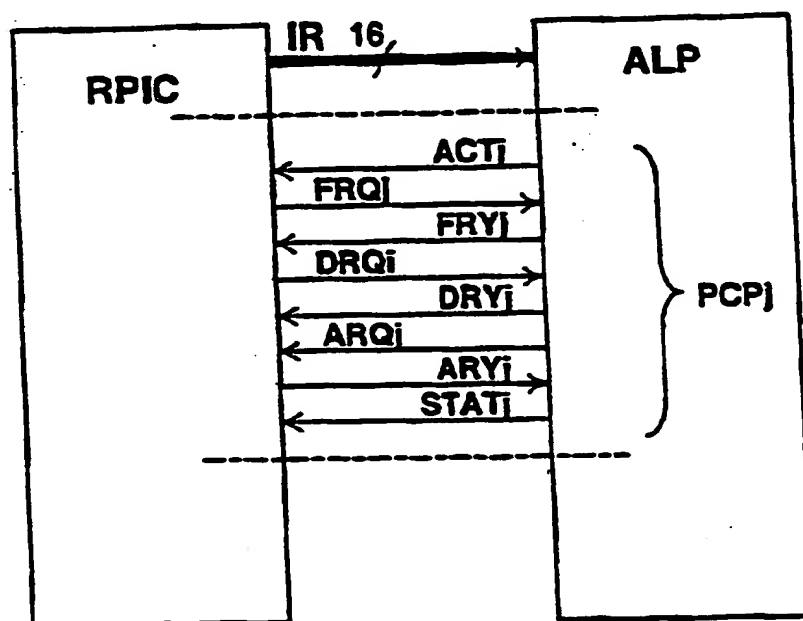


Fig. 8 Beispielhaftes Pipelinesegment

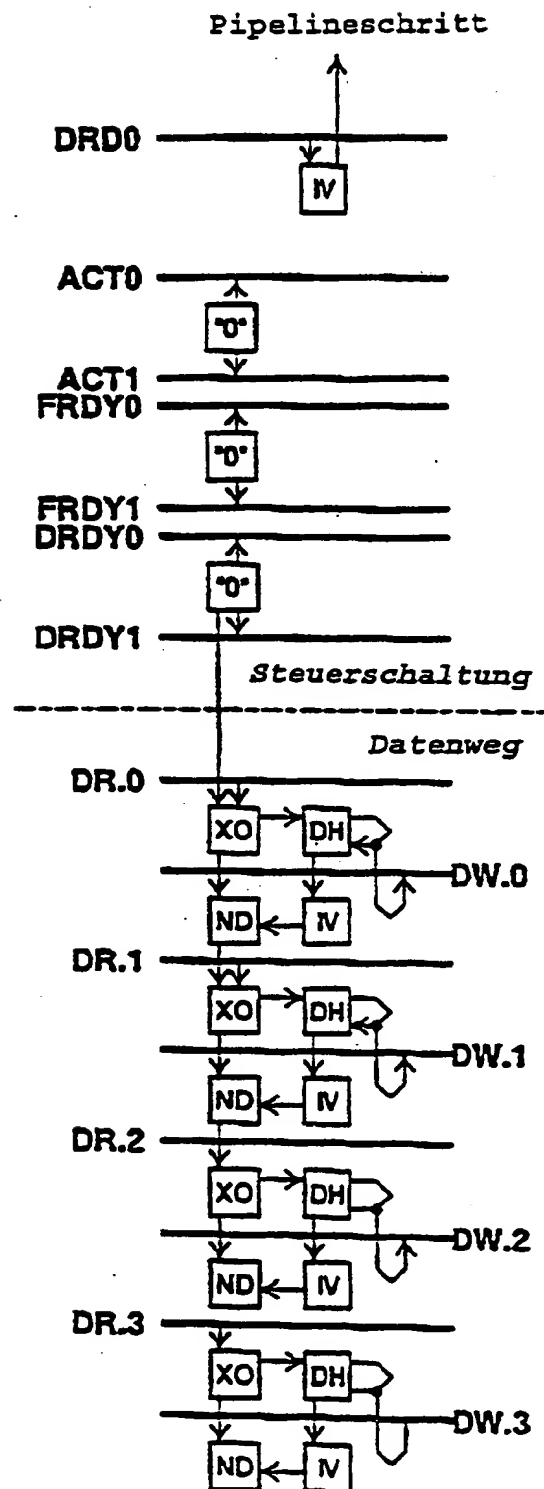


Fig. 9 Kernzellenschnittstelle

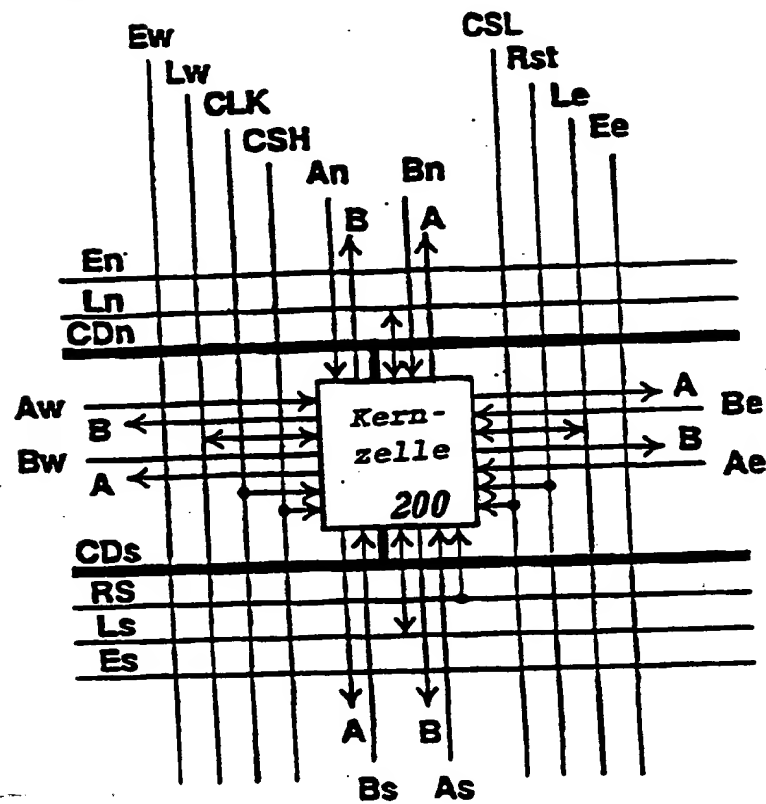


Fig. 10 ALP-Kernzellenschaltung

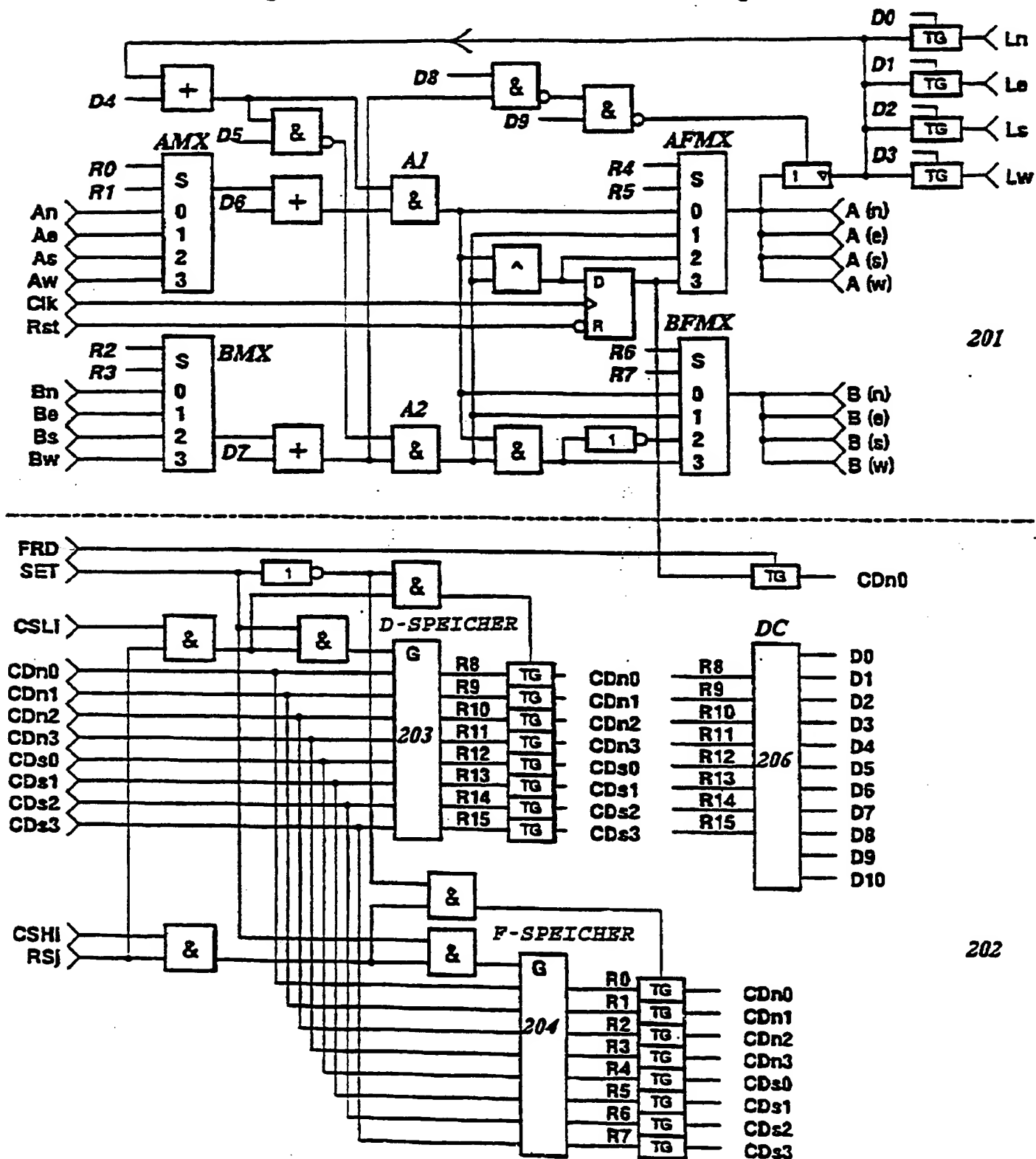


Fig. 11
Spaltentaktkonfigurationsschaltung

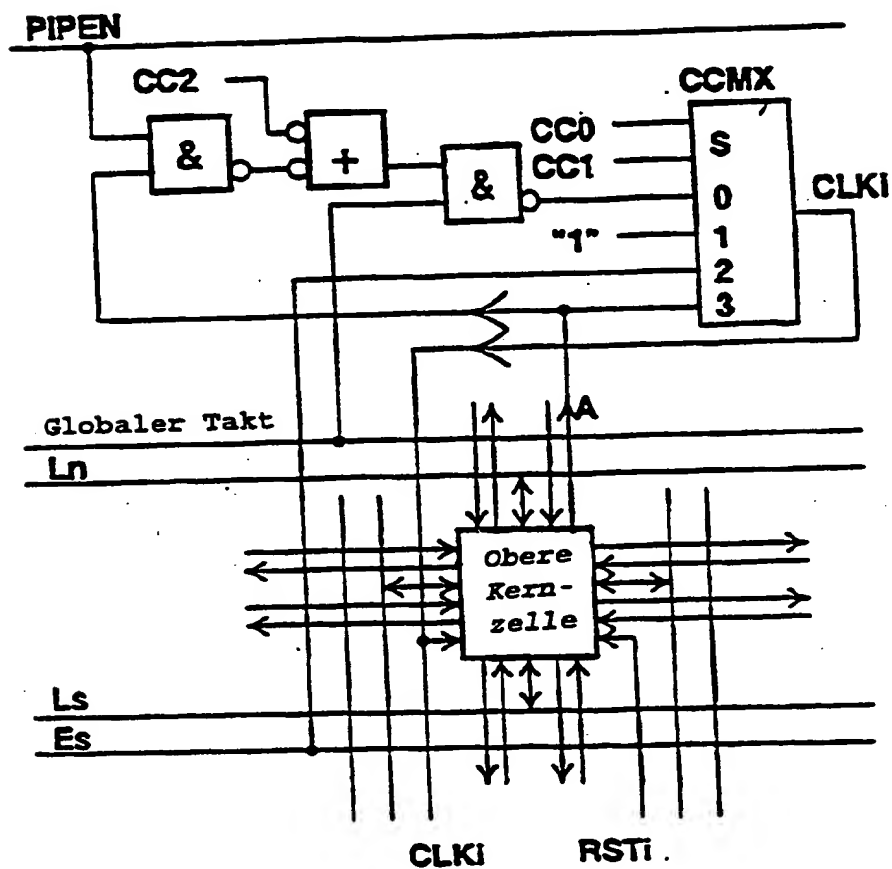


Fig. 12 Allgemeine Kernblockverbindungen

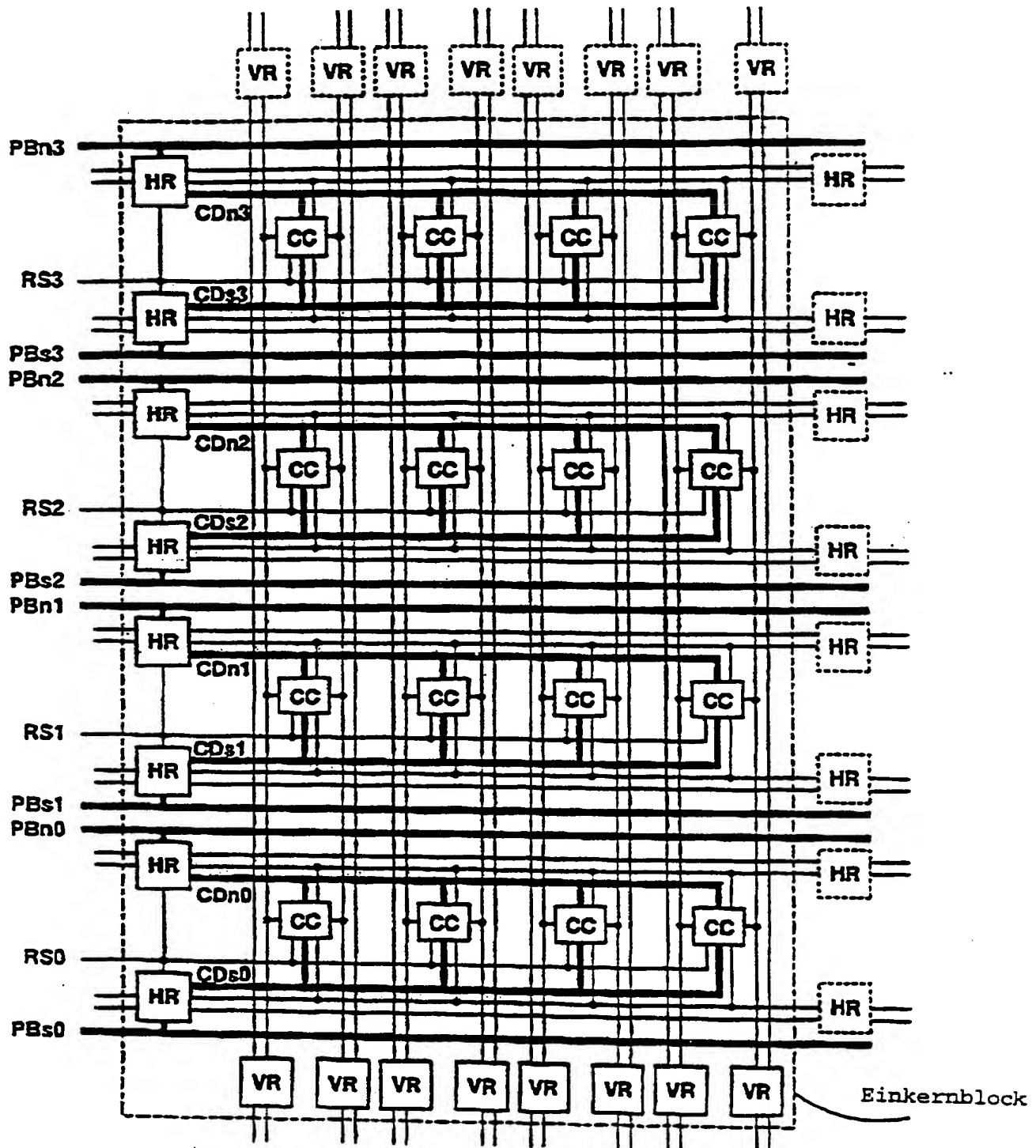


Fig. 13

Kernzellennachbar- und Verstärkerverbindungen

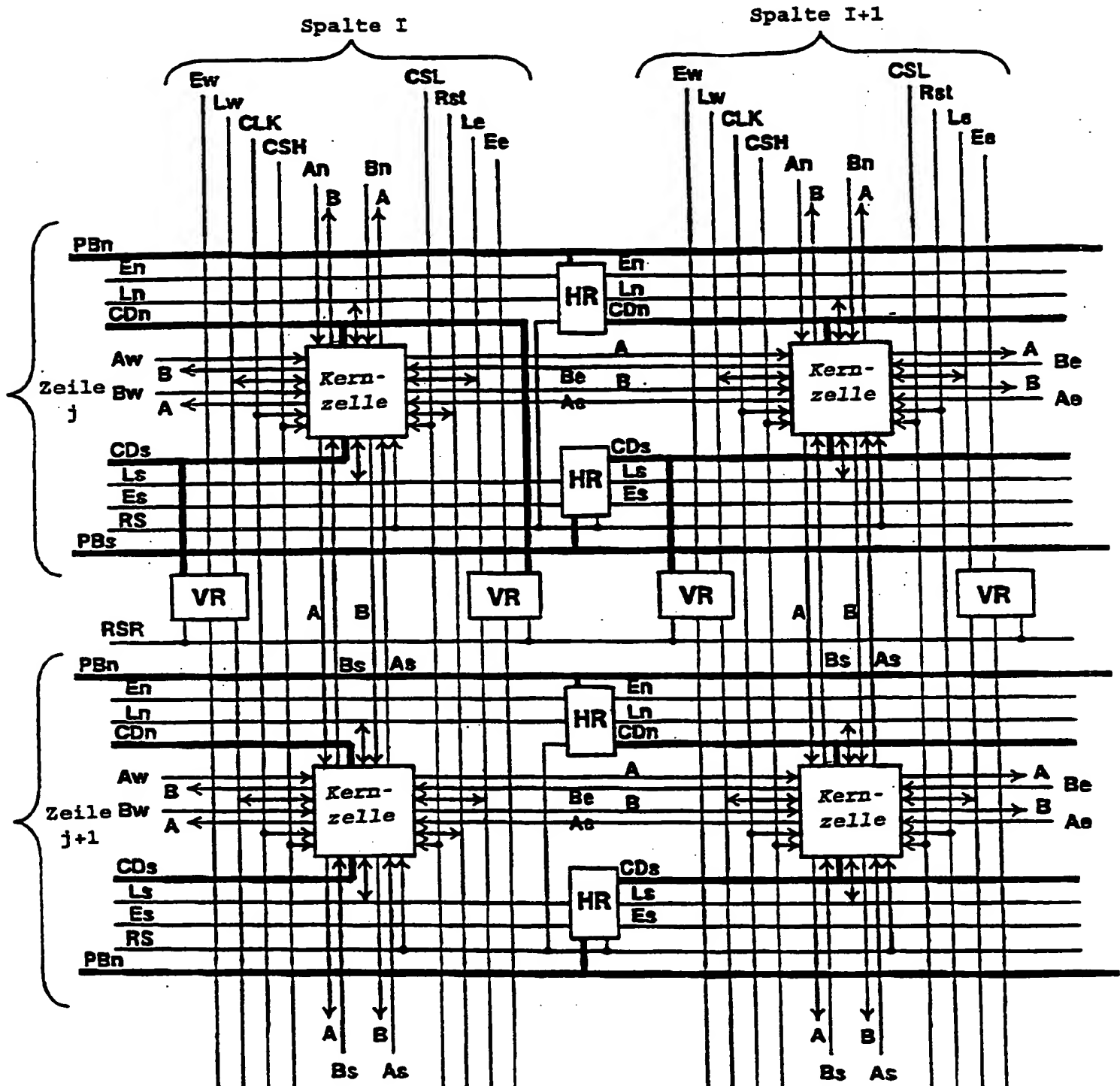


Fig. 14 Vertikale Verstärkerschaltung

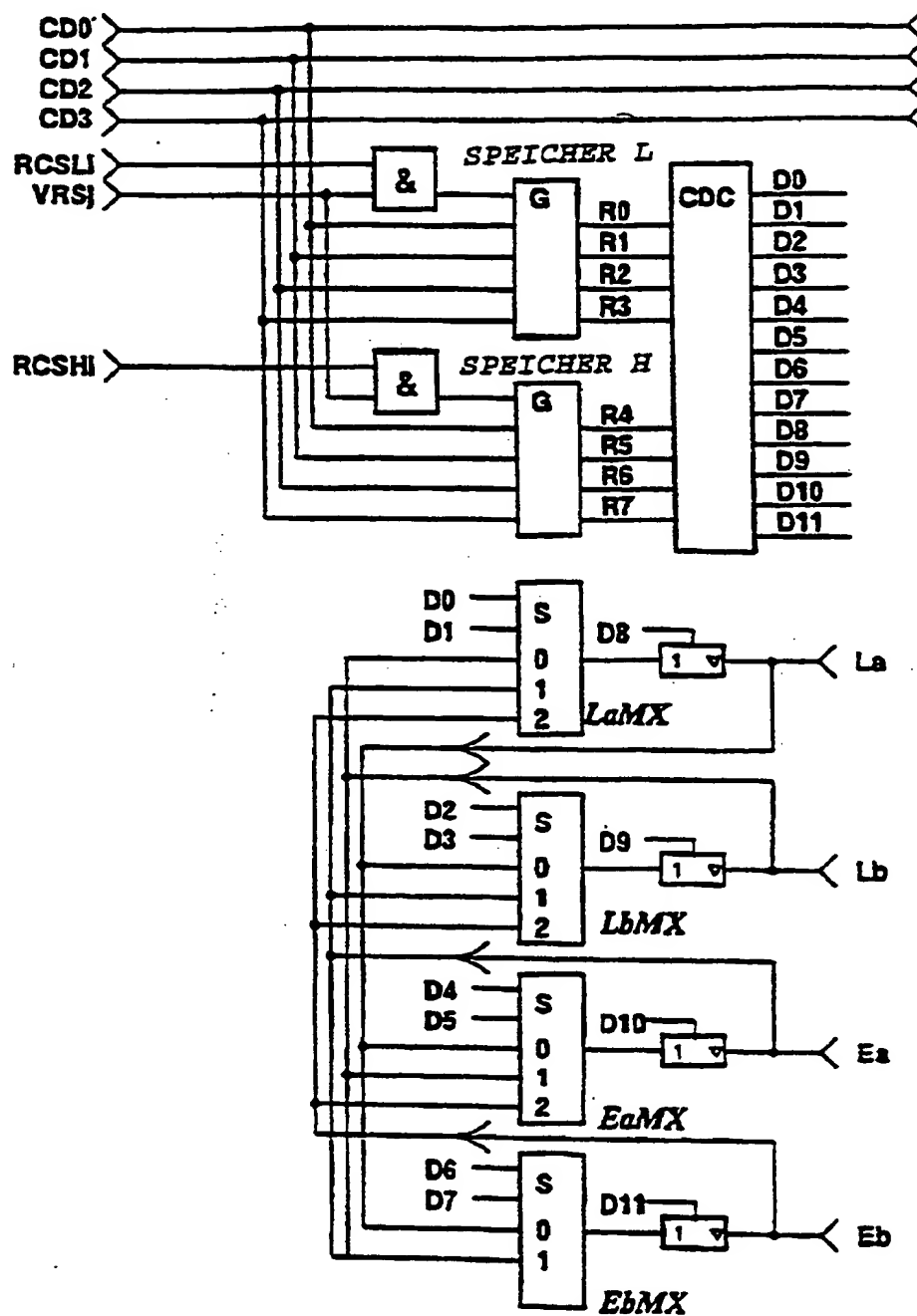


Fig. 15 Horizontale Verstärkerschaltung

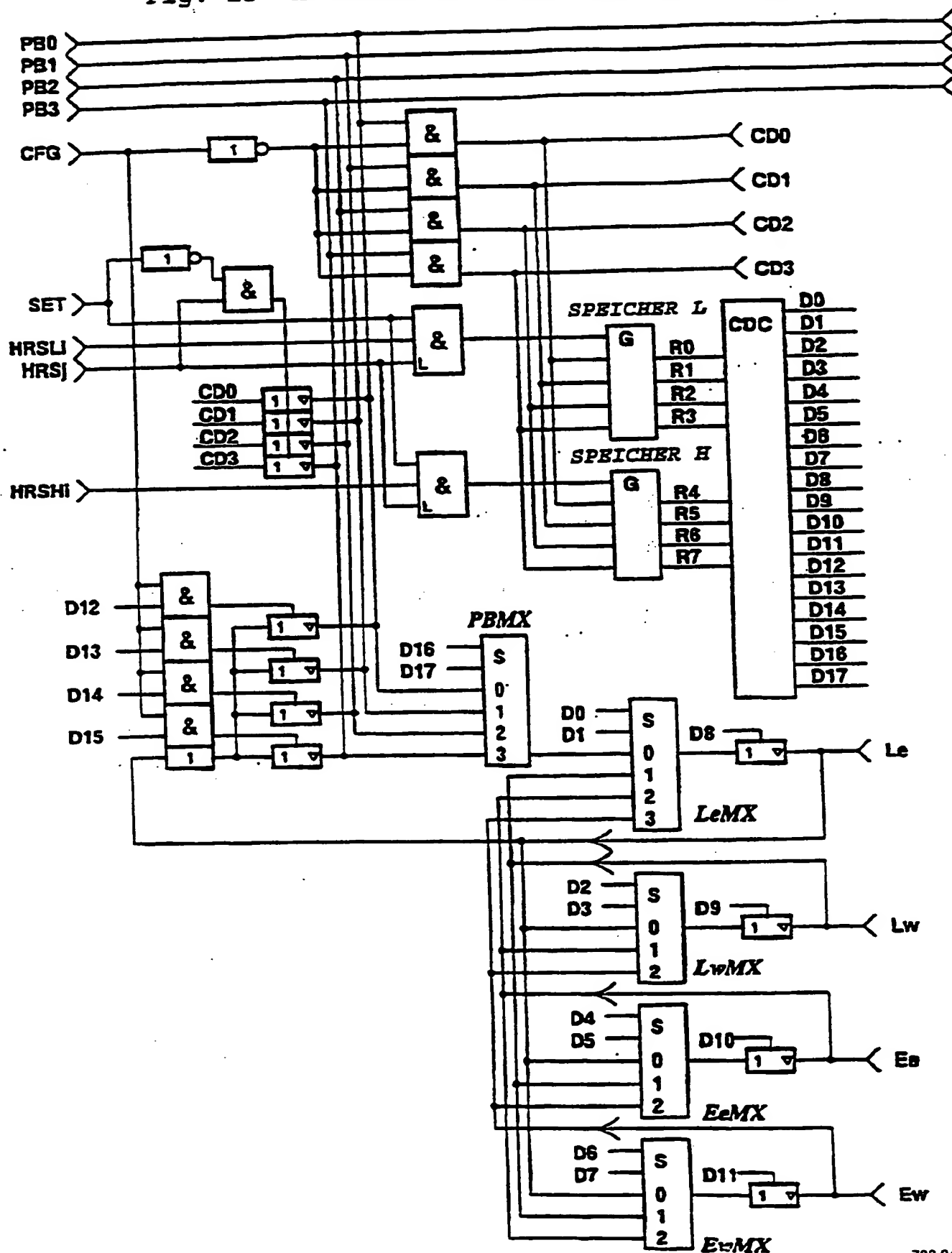


Fig. 18 RPIC-Grundblöcke

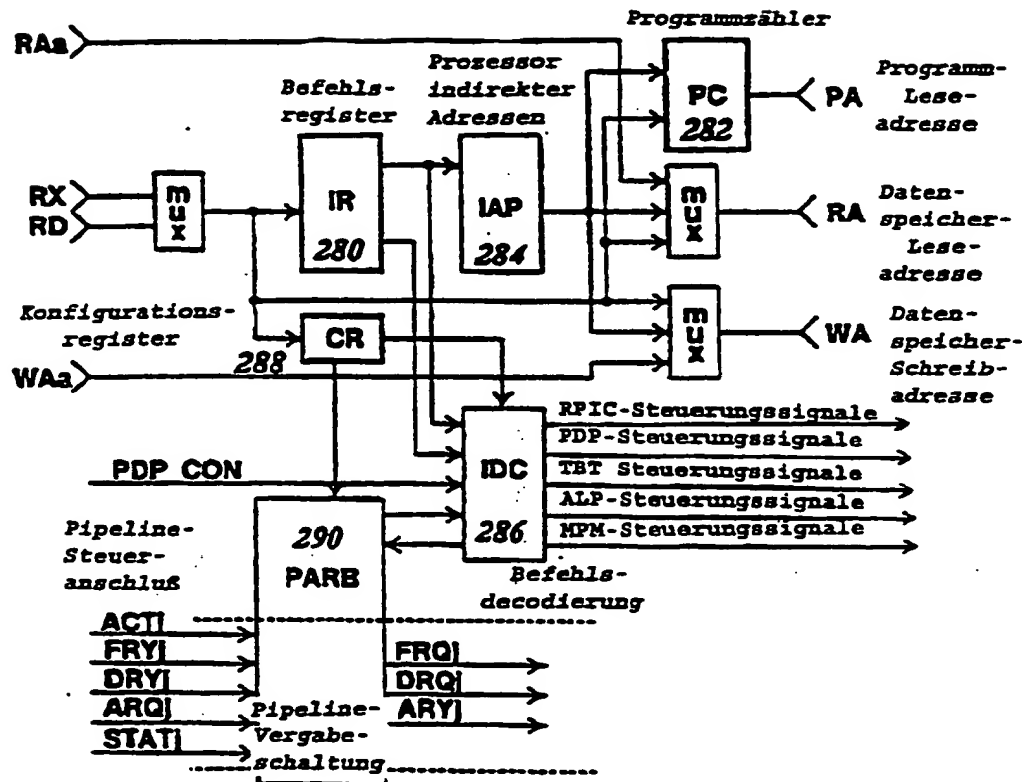


Fig. 19

Pipelinesteueranschluß-Freigabezustandsfluß

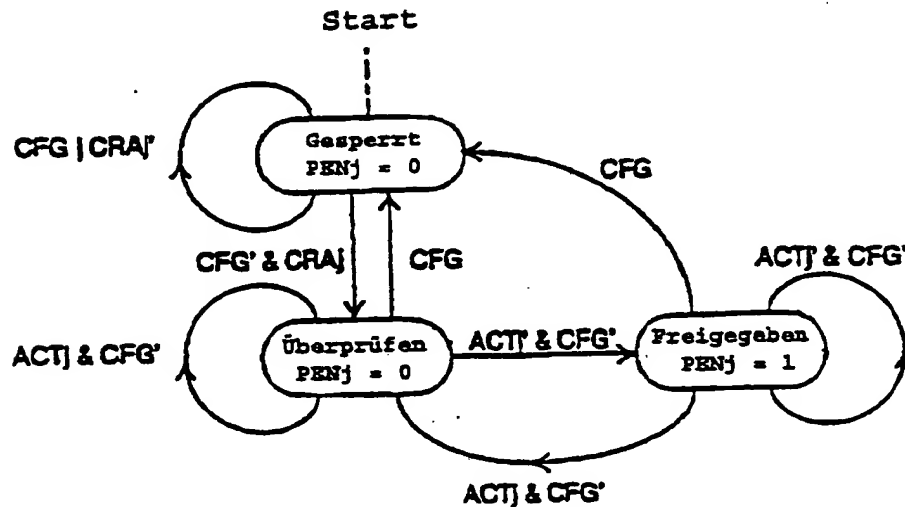


Fig. 20 Programmfunktionsanforderung

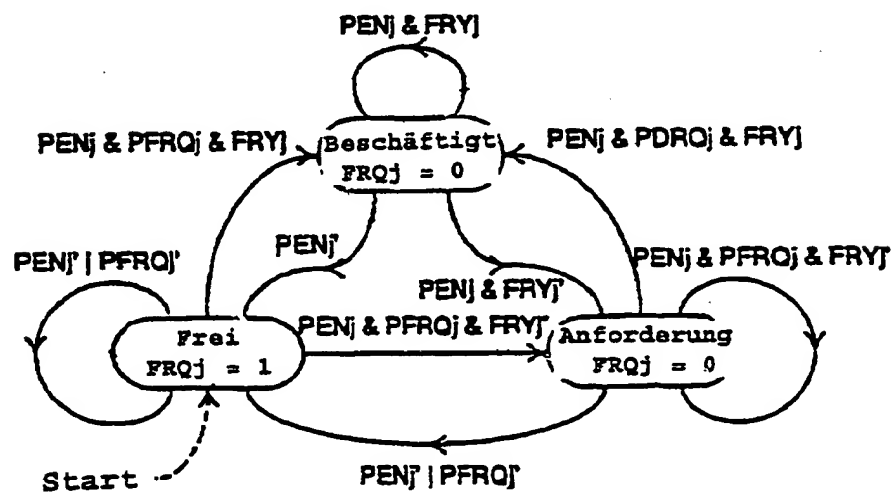


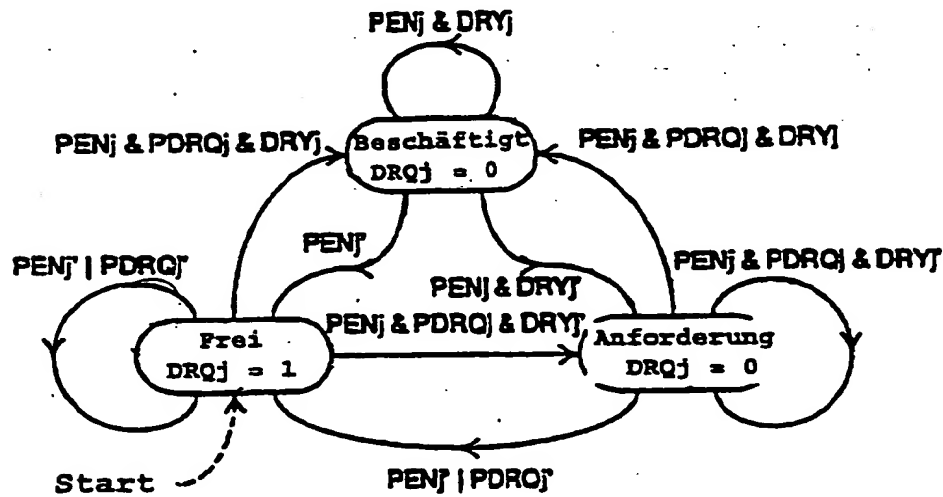
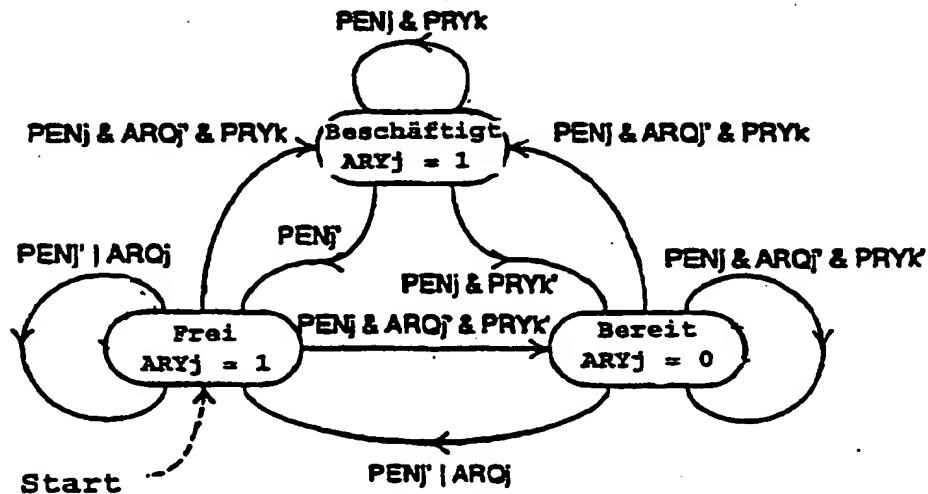
Fig. 21 Programmdatenanforderungszustandsfluß

Fig. 22
 Fluß des selbständigen Anforderungszustands


Fig. 23 IEEE-32-Bit-Gleitkommaformat

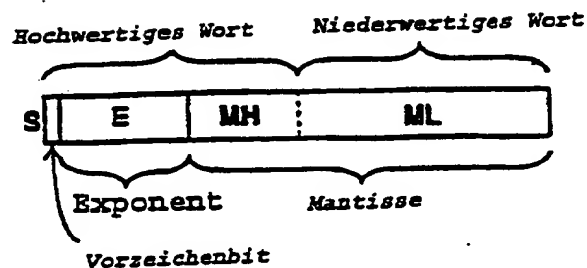


Fig. 24

ALP-Schaltung für Gleitkommaoperationen

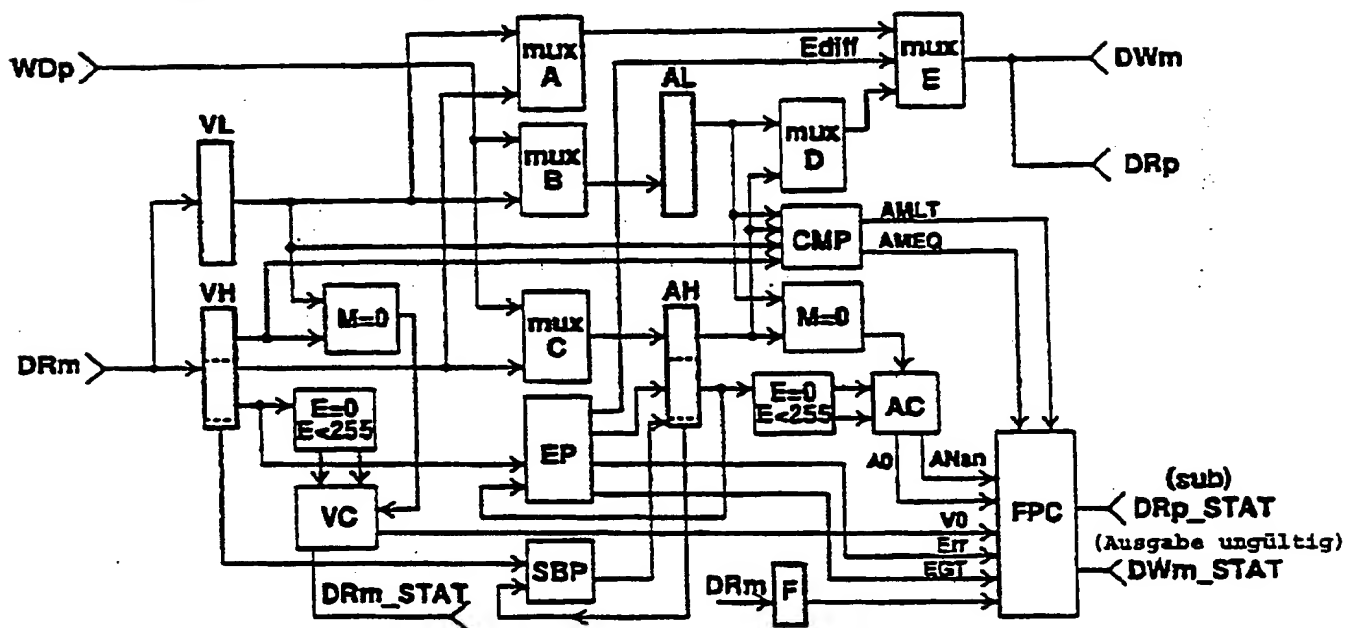


Fig. 25 Acht-Punkt-FFT-Datenfluß

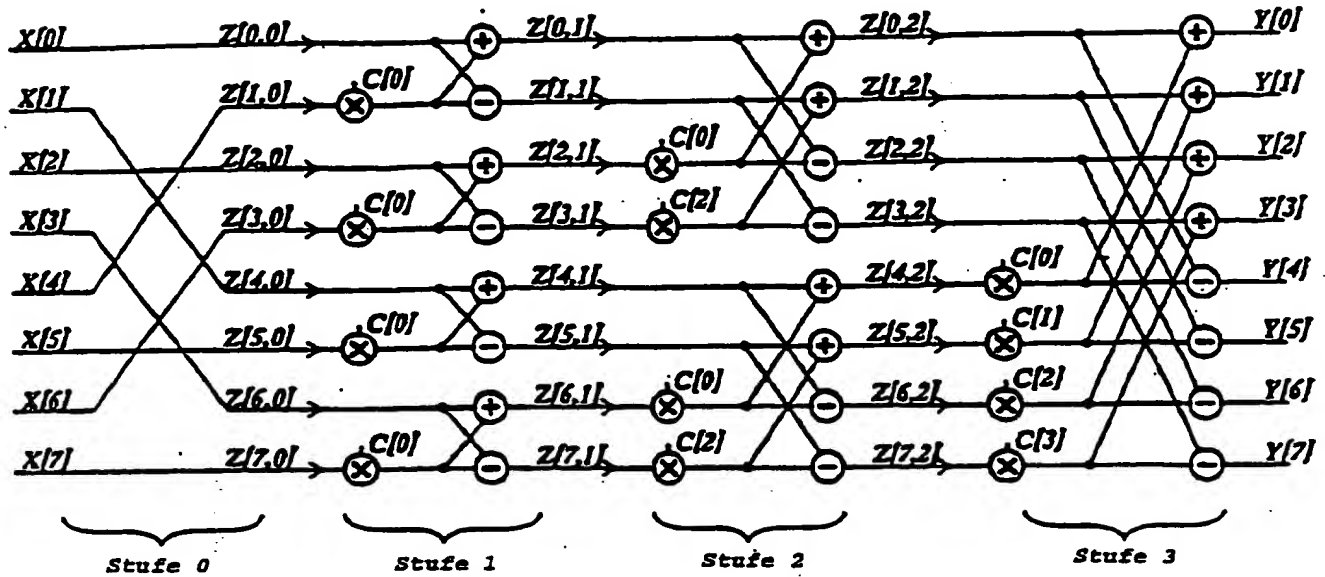


Fig. 26 ALP-Schaltung für Rückpermutation

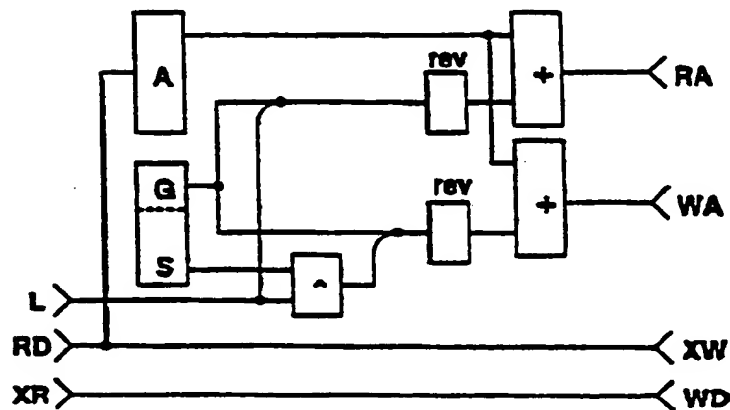


Fig. 27 FFT-Paar-Berechnung

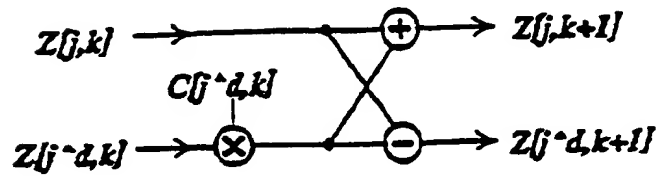


Fig. 28 ALP-Schaltung für FFT-Adreßerzeugung

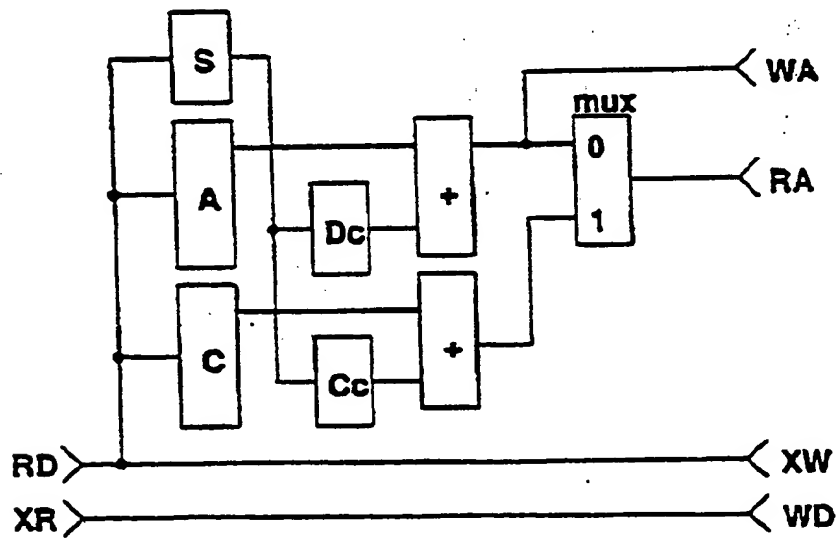


Fig. 29 Bildspeicherungshierarchie

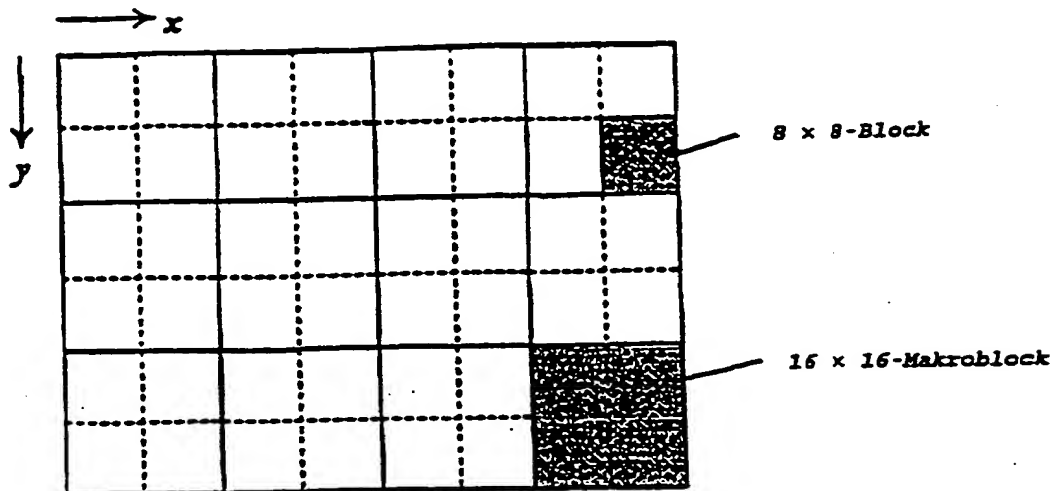


Fig. 30 Pel-Speicherplan

		→ x															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
↓ y	0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	1	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7
	2	4	5	6	7	0	1	2	3	C	D	E	F	8	9	A	B
	3	C	D	E	F	8	9	A	B	4	5	6	7	0	1	2	3
	4	2	3	0	1	6	7	4	5	A	B	8	9	E	F	C	D
	5	A	B	8	9	E	F	C	D	2	3	0	1	6	7	4	5
	6	6	7	4	5	2	3	0	1	E	F	C	D	A	B	8	9
	7	E	F	C	D	A	B	8	9	6	7	4	5	2	3	0	1
	8	1	0	3	2	5	4	7	6	9	8	B	A	D	C	F	E
	9	9	8	B	A	D	C	F	E	1	0	3	2	5	4	7	6
	A	5	4	7	6	1	0	3	2	D	C	F	E	9	8	B	A
	B	D	C	F	E	9	8	B	A	5	4	7	6	1	0	3	2
	C	3	2	1	0	7	6	5	4	B	A	9	8	F	E	D	C
	D	B	A	9	8	F	E	D	C	3	2	1	0	7	6	5	4
	E	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8
	F	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

THIS PAGE BLANK (USPTO)

Diagram illustrating a 3-stage interconnection network (Stufe 0, Stufe 1, Stufe 2, Stufe 3) with 8 inputs and 8 outputs. The nodes are labeled $Z[i,j]$ where i is the row index (0 to 7) and j is the stage index (0 to 3). The connections between stages are as follows:

- Stufe 0 to Stufe 1:**
 - $Z[0,0] \rightarrow Z[0,1]$ (+)
 - $Z[1,0] \rightarrow Z[1,1]$ (-)
 - $Z[2,0] \rightarrow Z[2,1]$ (+)
 - $Z[3,0] \rightarrow Z[3,1]$ (-)
 - $Z[4,0] \rightarrow Z[4,1]$ (+)
 - $Z[5,0] \rightarrow Z[5,1]$ (-)
 - $Z[6,0] \rightarrow Z[6,1]$ (+)
 - $Z[7,0] \rightarrow Z[7,1]$ (-)
- Stufe 1 to Stufe 2:**
 - $Z[0,1] \rightarrow Z[0,2]$ (+)
 - $Z[1,1] \rightarrow Z[1,2]$ (+)
 - $Z[2,1] \rightarrow Z[2,2]$ (-)
 - $Z[3,1] \rightarrow Z[3,2]$ (-)
 - $Z[4,1] \rightarrow Z[4,2]$ (+)
 - $Z[5,1] \rightarrow Z[5,2]$ (+)
 - $Z[6,1] \rightarrow Z[6,2]$ (-)
 - $Z[7,1] \rightarrow Z[7,2]$ (-)
- Stufe 2 to Stufe 3:**
 - $Z[0,2] \rightarrow Z[0,3]$ (+)
 - $Z[1,2] \rightarrow Z[1,3]$ (+)
 - $Z[2,2] \rightarrow Z[2,3]$ (+)
 - $Z[3,2] \rightarrow Z[3,3]$ (+)
 - $Z[4,2] \rightarrow Z[4,3]$ (-)
 - $Z[5,2] \rightarrow Z[5,3]$ (-)
 - $Z[6,2] \rightarrow Z[6,3]$ (-)
 - $Z[7,2] \rightarrow Z[7,3]$ (-)

```

graph TD
    XR --> Mux
    RD --> Mux
    Mux --> AddSub[Addition/  
Subtraktion]
    sub --> AddSub
    AddSub --> Takt
    Takt --> XW
    Takt --> WD
    Takt --> AddSub
  
```

The diagram illustrates the ALU (Arithmetic Logic Unit) structure. It features two input registers, XR and RD, which feed into a Multiplexer (Mux). The output of the Mux is directed to the Addition/Subtraktion block. A control signal, labeled 'sub', is also provided to the Addition/Subtraktion block. The output of the Addition/Subtraktion block is then sent to the Takt (Clock) block. The Takt block has two outputs, XW and WD, which are fed back into the Addition/Subtraktion block. Additionally, the Takt block has a direct output path to the Addition/Subtraktion block.

DOCKET NO: GR98P8107

SERIAL NO: 09/816,926

APPLICANT: Arnold et al.

LERNER AND GREENBERG P.A.

P.O. BOX 2480

HOLLYWOOD, FLORIDA 33022

TEL. (954) 925-1100